Book Usage Data Workflows

Curtin University

Nov 08, 2023

CONTENTS

Documentation on how the Dashboard works and information on the data sources used.

CHAPTER

ONE

DASHBOARD OVERVIEW

1.1 Book Analytics Dashboard

1.1.1 How can the Book Analytics Dashboard help me?

Whether you're a publisher, librarian, funder, administrator, or other stakeholder in the scholarly communications community, the Dashboard can help you gain a fuller view of book usage data.

Books are made available through a multitude of different platforms, and each has its own way of providing usage statistics. Using our Dashboard gives you a single point of synthesis of usage data from a wide range of sources, allowing you a consolidated view of usage.

We take care to bring together a synthesis of usage data of known provenance to allow you to compare data across your chosen time span. Learn more about how we collect and process book usage data.

1.1.2 Who's behind the Dashboard and how is it operated?

The Dashboard was initially developed as a pilot project 2020-2022, which developed a prototype for gathering book usage information from multiple data sources, and combining and presenting it in interactive visualisation dashboards for publisher partners.

Funded again by the Mellon Foundation, the pilot project was scaled up to become the Book Analytics Dashboard project (2022-2025), focused on creating a sustainable OA book focused analytics service.

The fully-functional Dashboard is now operated by OAPEN, a trusted infrastructure for OA books. OAPEN is a not-for-profit organisation which cannot be sold, and has committed to the Principles of Open Scholarly Infrastructure (POSI) with a public self-audit of their practices across the themes of governance, sustainability, and insurance. OAPEN was chosen as the host organisation by the Book Analytics Dashboard (BAD) project Advisory Board, following a series of focus groups with publishers which endorsed OAPEN as an organisation which could be trusted by the community to run the Dashboard in an open and consultative way.

Community voices are an essential part of the governance strategy for the BAD project. We want to ensure that effective channels are available for feedback, feature requests, praise, and suggestions for improvement; and that these are regularly reviewed and connected to our development planning. During Q3 and Q4 2023, we will be working directly with Dashboard partners to imagine, design, and implement community engagement processes that will allow easy communication and proactive connectivity between the BAD project team and community members.

1.2 How the Dashboard works

1.2.1 How does the Dashboard collect and process book usage data?

At the heart of the Dashboard's technology stack is a valid ONIX feed which includes the metadata of the works a partner wishes to represent in the dashboard. We use ONIX because this is the book industry's standard metadata interchange format that publishers use to share information about the books that they have published.

Our workflows collect book usage data from multiple sources (learn more about our data sources) and public bibliographic metadata and Event Data from Crossref. Data from these sources is integrated with the ONIX feed, using the ISBN-13 identifier to identify works and combine usage data from multiple sources. The partner's ONIX feed serves as the source of truth for a work's metadata, such as book title, authors, and related works. Crossref bibliographic metadata is used to match Event DOIs with book ISBNs, which are then matched with the ISBNs in the partner's ONIX feed.

Our workflows are the code which controls the data integration; all of which is built on an open-source workflow system. The workflows fetch, process, disambiguate, and analyse data about books from multiple sources, and this data is saved to Google Cloud's BigQuery data warehouse. The next steps of data processing include:

1. Ingesting data via telescope workflows from Crossref metadata, Crossref Event Data, Google Analytics, Google Books, JSTOR, IRUS Fulcrum, IRUS OAPEN, a publisher's ONIX feed (obtained via SFTP, or from the OAPEN Library, or from Thoth), UCL Discovery, and
2. A series of analytic workflows to process and combine the data ingested by the telescope workflows. The processed data in the Google Cloud BigQuery data warehouse is then visualised in dashboards provided by Looker Studio, a dashboarding solution offered by Google.

The information from our data sources is refreshed on a regular basis, keeping the Dashboard up-to-date. Updated usage data for all sources is available on the dashboard typically on the first Monday after the fourth of the month. Crossref Event data is typically updated weekly.

Is the Dashboard data COUNTER-conformant?

Please see our Dashboard data sources overview which gives you a detailed overview of each source.

How do we deal with bot activity?

Bot identification is the responsibility of the platforms themselves, as they have access to the individual usage data, which we do not. Platforms that are using COUNTER-conformant standards (such as IRUS OAPEN usage statistics) should only include genuine, user-driven usage, as activity generated by internet robots and crawlers must be excluded from all COUNTER usage reports.

1.2.2 How is the Dashboard data protected?

We receive usage data from platform providers in aggregated and anonymised format: individual usage data is stripped out so that the data we receive is an aggregation and can't be traced back to individuals. In the event that platform usage reports contain location information such as individual IP address, this information is anonymised before it is provided to the Dashboard. For example, IRUS OAPEN usage reports do contain IP addresses, therefore this data is downloaded and anonymised within an OAPEN Google Cloud project located in Europe. The transformed data, with IP addresses removed and replaced with city or country information, is then sent to the Dashboard.

Since we do not collect any personally identifiable information, GDPR does not apply to our data.

Each partner's data is kept in a separate Google Cloud project (located in the USA). Access to each is controlled with user access permissions (username and password credentials), providing strong security and privacy. Only Dashboard staff have access to this partner data.

Data provided is used only for the purposes of the Dashboards: it is not sold on or made available to any other parties for any reason.

1.3 Dashboard data sources

1.3.1 What are the Dashboard's data sources?

To see the data sources for a specific Dashboard, click on About & FAQ on the Dashboard, and consult the list at Data Sources. The only obligatory data source is title metadata in ONIX format; each publisher then chooses the other data sources they wish to include.

The data sources currently available to be visualised in the Dashboard are detailed in the tables below. The standard data sources and variables used are included, other data sources and variables may be supported as an extra add-on service.

Where the Dashboard gets title metadata from:

Data source	Status	Access
Crossref metadata	Current	Public
OAPEN metadata	Current	Public
ONIX-FTP feed from publishers	Current	Private
Thoth	Current	Public

Where the Dashboard gets usage and event data from:

Data source	Status	Access	COUNTER conformant?	Page Views
Crossref Event Data	Current	Public	n/a	
Google Analytics Universal	Not current	Private	No	Y [page_views]
Google Books	Current	Private	No	
IRUS Fulcrum	Current	Private	Yes	
IRUS OAPEN	Current	Private	Yes	
JSTOR	Current	Private	Yes	
UCL Discovery	Current	Public	No	

… continued

Data source	Book Views	Book Downloads	Chapter Downloads	Time aggregation
Crossref Event Data				Monthly

continues on next page

Table 3 – continued from previous page

Data source	Book Views	Book Downloads	Chapter Downloads	Time aggregation
Google Analytics Universal		Y (with custom dimensions)		Monthly
Google Books	Y [BV_with_Pages_Viewed]	Y [qty]		Monthly
IRUS Fulcrum		Y [title_requests] and [total_item_requests]		Monthly
IRUS OAPEN		Y [title_requests] and [total_item_requests]		Monthly
JSTOR			Y [total_item_requests]	Monthly
UCL Discovery		Y [total_downloads]		Monthly

1.3.2 Public access data sources

The public access data sources are those where data is made publicly available by the data source. No additional access permission is required from Dashboard partners for the Dashboard to access the following data sources if partners want them to be included on their dashboard/s.

Crossref Event Data

Crossref Event Data captures online discussion about research outputs, such as 'a citation in a dataset or patent, a mention in a news article, Wikipedia page or on a blog, or discussion and comment on social media'. The event data is retrieved using the Crossref Events API. Crossref Event data must be queried using a DOI, which the Dashboard obtains from Crossref Metadata.

Crossref metadata

Crossref is a non-for-profit membership organization, and an official Digital Object Identifier (DOI) Registration Agency of the International DOI Foundation. They make metadata available for all DOIs registered with Crossref. The Dashboard uses Crossref Metadata to match ISBNs obtained from a publishers Onix feed to DOIs to query Crossref Event Data.

OAPEN metadata

OAPEN enables libraries and aggregators to use the metadata of all available titles in the OAPEN Library. The metadata is available in different formats and the Dashboard harvests the data in the XML format to obtain an file in ONIX format for the OAPEN platform.

Thoth

Thoth is a free, open metadata service that publishers can choose to utilise as a solution for metadata storage. Thoth can provide metadata upon request in a number of formats. The Dashboard uses the Thoth Export API to download metadata for publishers in an ONIX format.

UCL Discovery

University College London (UCL) is an eBook publisher, and partner in the Dashboard. UCL Discovery is UCL's open access repository, showcasing and providing access to the full texts of UCL research publications.

1.3.3 Private access data sources - access permission required

The following private access data sources require specific access permissions to be granted from the Dashboard partner for the Dashboard to access data if partners want them to be included on their dashboard/s.

Google Analytics Universal

Google Analytics Universal monitors and records web traffic for specific websites. If a Dashboard partner had configured Google Analytics on their publisher website, the Google Analytics data can be used to find out which countries and territories website visitors are from.

Google Books

The Google Books Partner program hosts eBooks, including some free open access eBooks. eBook publishers can then download usage reports from Google Books. The Dashboard uses data from the Google Play sales transaction report and the Google Books Traffic Report.

JSTOR

JSTOR is a digital library, offering over 7000 open access eBooks. Publisher usage reports offer details about the use (views and downloads) of eBooks by institution, and country.

ONIX-FTP feed from publishers

ONIX is a standard that book publishers use to share information about the books that they have published. BAD project dashboard partners that have ONIX feeds are given credentials and access to their own upload folder on the Mellon SFTP server. The BAD project dashboard partner uploads their ONIX feed to their upload folder on a weekly, fortnightly or monthly basis. The Book Usage Data Workflows ONIX telescope downloads, transforms (with the ONIX parser Java command line tool) and then loads the ONIX data into BigQuery for further processing.

1.3.4 Private access data sources - no additional access permission required

The following private access data sources are already available to the Dashboard through existing arrangements. They do not require additional access permissions to be granted from the Dashboard partners for the Dashboard to access data if partners want them to be included on their dashboard/s.

IRUS Fulcrum

IRUS provides COUNTER standard access reports for eBooks hosted on the Fulcrum platform. Fulcrum is a "community-developed, open source platform for digital scholarship" which provides "users the ability to read books with associated digital enhancements, such as: 3-D models, embedded audio, video, and databases; zoomable online images, and interactive media".

IRUS OAPEN

IRUS provides COUNTER standard access reports for eBooks hosted on the OAPEN library and platform. OAPEN "promotes and supports the transition to open access for academic books by providing open infrastructure services to stakeholders in scholarly communication". Almost all eBooks on OAPEN are provided as a PDF file for the whole book. The reports show access figures for each month, and the location (IP address) of the access. Within the OAPEN Google Cloud project (located in Europe), IP addresses are replaced with geographical information (city and country). This means that IP addresses are not stored within the Dashboard data, and only de-identified geographical information transferred to the Dashboard.

1.4 Dashboard use and FAQ

1.4.1 What can I see in the Dashboard?

Publishers, in your Dashboard, you can see usage data for your own published works. You can see the usage of the books you have published in terms of views, downloads, and online mentions and events. You can also view which countries and institutions are using your books, and which subjects are represented in your collections.

Features to explore:

- **Reset** - use the "Reset" button at the top of the page, or "Reset filters" to the right of the filters selection
- Search by **author name** (currently displayed as last name, first name) learn more in ORCID's documentation about first and last names)
- **Track usage over time** - from the "Overview" page, see which month has the highest usage numbers when all book titles are selected
- See **usage for a specific title** - from the "Overview" page, select a book title from the filter and see the total access number
- See **usage across geographic regions** - from the "Global Reach" page, select a country from the country filter and see on the number of book downloads and the number of chapter downloads. Countries and territories are based on ISO standard 3166.

1.4.2 I'm not yet participating, but can I see a demo?

You may view the BAD template Dashboard, powered by usage data from the University of Michigan Press which they have very kindly made publicly available, with no login details required.

1.4.3 How should I interpret the data from the Dashboard?

Presenting a holistic view of usage information can be difficult, because ebooks can be hosted in multiple repositories and platforms, in different file formats (PDF, EPUB, MOBI, HTML), and in different levels (whole book or by chapter). Each repository provides book content to different audiences in different ways.

Before studying data from the Dashboard, we suggest that you take some time to understand the data sources, including the limitations of comparing different data sources. Learn more about how usage is influenced by the language of the work, its subject, its platform, and seasonal differences in Ronald's blog post and research paper, "Measured in a context: making sense of open access book data".

Caveats:

- For publishers with a smaller number of titles, it's harder to see a pattern and understand why some get more downloads than others
- Incomplete data can hamper making accurate comparisons

But all that said, the Dashboard can help you explore some interesting questions!

Publishers, here are some questions the Dashboard data can help you consider:

- In which countries and territories are my publications most and least downloaded?
- Does this correspond to the languages in which I'm publishing?
- Which subjects are most or least popular in different areas?
- How does this change over time?

CHAPTER

TWO

MORE INFORMATION

2.1 More information and contact us

For all your enquiries and questions, please contact us at info@book-analytics.org.

Book Analytics Service key information page, including how to get started

Information about the Mellon Foundation funded Book Analytics Dashboard (BAD) project (2022-2025)

- Visit the BAD project website
- Follow the BAD project on Twitter @BookAnalytics
- Join the BAD project mailing list
- Visit the BAD project Zenodo community
- See the BAD template dashboard, powered by the University of Michigan Press' data
- Our book-focussed GitHub repository

Running in parallel with the Book Analytics Dashboard project is the OA Book Usage Data Trust, a project to formalise community governance mechanisms, quantify data trust participation benefits, and understand the full operational costs related to an international data space for OA book usage.

2.2 Glossary

BAD
: The Book Analytics Dashboard Project (2022-2025) - a term used to refer to the Mellon Foundation funded project that is focused on creating a sustainable open access Book focused analytics service[1]

COKI
: Curtin Open Knowledge Initiative - a team of data scientists, software developers and researchers at Curtin University, Perth, Australia[2]

COUNTER
: COUNTER provides the standard that enables the knowledge community to count the use of electronic resources. To have their usage statistics and reports designated COUNTER compliant, report providers MUST provide usage statistics that conform to the current Code of Practice[3]

[1] https://openknowledge.community/projects/bad-project/
[2] https://openknowledge.community/
[3] https://www.projectcounter.org/

Crossref
Crossref is a Digital Object Identifier (DOI) Registration Agency of the International DOI Foundation, that makes metadata available for all DOIs registered with them[4]

dashboard
A dashboard is an interactive, up-to-date page of visualisations that aggregate and summarise data from different sources

data source
A public or pilot project dashboard partner source of data about open access eBooks and their usage, such as views, downloads and online mentions

eBook
A book publication made available in electronic or digital form

Google Books
Google Books provides paid and free (open access) eBooks[5]

IRUS
IRUS – a service for capturing and processing institutional repository usage data, making it possible for institutional repositories and platforms to generate COUNTER compliant usage data[6]

IRUS Fulcrum
Fulcrum is a "community-developed, open source platform for digital scholarship". IRUS provides COUNTER standard usgae reports for eBooks hosted on the Fulcrum platform[7]

IRUS OAPEN
IRUS provides COUNTER standard usage reports for eBooks hosted on the OAPEN library and platform[8]

JSTOR
JSTOR is a digital library, which offers over 7000 open access eBooks[9]

Looker Studio
A dashboarding solution provided by Google[10]

OAeBU
Open Access eBook Usage (2020 - 2022) - a term used to refer to the Mellon Foundation funded pilot project Developing a Pilot Data Trust for Open Access Ebook Usage[11]

OAPEN
OAPEN is a not-for-profit organisation dedicated to open access, peer-reviewed books, operating three platforms: OAPEN Library; OAPEN Open Access Books Toolkit; and Directory of Open Access Books[12]

open access
Open access (OA) is free access to information, and unrestricted use of electronic resources for all[13]

ONIX
ONIX for Books (ONline Information eXchange) is a standard format that book publishers use to share information about the books that they have published[14]

[4] https://www.crossref.org/community/
[5] https://play.google.com/books/publish/
[6] https://www.jisc.ac.uk/irus
[7] https://www.fulcrum.org/
[8] https://www.oapen.org/
[9] https://about.jstor.org/librarians/books/open-access-books-jstor/
[10] https://cloud.google.com/looker-studio
[11] https://educopia.org/data_trust/
[12] https://www.oapen.org/oapen/1891940-organisation
[13] https://en.unesco.org/open-access/what-open-access
[14] https://bisg.org/general/custom.asp?page=ONIXforBooks

SFTP
SSH File Transfer Protocol

telescope
A telescope is a data workflow that fetches and ingests data from a data source. Some telescopes run workflows that process and output data to other places. Workflows are built on top of Apache Airflow's Directed Acyclic Graph (DAGs), where a DAG is "a collection of organized tasks that you want to schedule and run"[15]

the Dashboard
Refers to the books analytics service operated by OAPEN[16]

[15] https://cloud.google.com/composer/docs/run-apache-airflow-dag
[16] https://oapen.org/article/book-analytics-service

CHAPTER

THREE

BOOK USAGE DATA WORKFLOWS

Documentation about the code/files hosted in the Book Usage Data Worfklows Github repository. This includes (technical) documentation on the telescope/analytical workflows that are a part of this repository, license info & contributing guidelines and auto-generated API reference documentation.

3.1 Book Usage Data Workflows

Book Usage Data Workflows provides Apache Airflow workflows for fetching, processing and analysing data about Open Access Books.

The workflows include: Google Analytics, Google Books, JSTOR, IRUS Fulcrum, IRUS OAPEN, ONIX, Thoth, UCL Discovery and an Onix Workflow for combining all of this data.

3.1.1 Telescope workflows

A telescope is a type of workflow used to ingest data from different data sources, and to run workflows that process and output data to other places. Workflows are built on top of Apache Airflow's DAGs.

Telescope workflows

A telescope is a type of workflow used to ingest data from different data sources, and to run workflows that process and output data to other places. Workflows are built on top of Apache Airflow's DAGs.

Crossref Events

> When someone links their data online, or mentions research on a social media site, we capture that event and make it available for anyone to use in their own way. We provide the unprocessed data—you decide how to use it.

Before the expansion of the Internet, most discussion about scholarly content stayed within scholarly content, with articles citing each other. With the growth of online platforms for discussion, publication and social media, we have seen discussions extend into new, non-traditional venues. Crossref Event Data captures this activity and acts as a hub for the storage and distribution of this data. An event may be a citation in a dataset or patent, a mention in a news article, Wikipedia page or on a blog, or discussion and comment on social media.

When someone links their data online, or mentions research on, for example, Twitter, Wikipedia, or Reddit, Crossref's uses a set of APIs to captures and records those events in their 'Event dataset'. Events are tracked via their DOI and URLs, which enables Crossref to monitor where it's been shared, linked, bookmarked, referenced or commented on. Crossref Event Data currently contains events from a range of data sources, including Crossref Metadata, DataCite

Metadata, F1000Prime (Recommendations of research publications, Hypothes.is, The Lens (Cambia), Newsfeed, Reddit, Reddit Links, Stack Exchange Network, Twitter, Wikipedia, and Wordpress.com

See the crossref events page, and data details, for more information.

The corresponding table created in BigQuery is `crossref.crossref_eventsYYYYMMDD`. This table is created during the Onix workflow.

Summary	
Average runtime	2 hours
Average download size	10 GB
Harvest Type	API
Harvest Frequency	Weekly
Runs on remote worker	True
Catchup missed runs	False
Table Write Disposition	Append
Update Frequency	Daily
Credentials Required	No
Uses Telescope Template	Stream

Latest schema

name	type	mode	description
id	STRING	REQUIRED	Unique ID for the Event.
subj_id	STRING	NULLABLE	Subject persistent ID.
relation_type_id	STRING	NULLABLE	Type of the relationship between the subject and object.
obj_id	STRING	NULLABLE	Object persistent ID.
timestamp	TIMESTAMP	REQUIRED	Timestamp of when the Event was created.
occurred_at	TIMESTAMP	REQUIRED	Timestamp of when the Event is reported to have occurred.
experimental	BOOL	NULLABLE	
total	INTEGER	NULLABLE	
source_id	STRING	REQUIRED	A name for the source.
source_token	STRING	NULLABLE	Unique ID that identifies the Agent that generated the Event.
terms	STRING	NULLABLE	Terms of use for using the API at the point that you acquire the Event.
license	STRING	NULLABLE	A license under which the Event is made available.
evidence_record	STRING	NULLABLE	Link to an Evidence Record for this Event.
subj	RECORD	NULLABLE	Subject metadata.
subj.pid	STRING	NULLABLE	The persistent ID. Must correspond to 'subj_id' or 'obj_id'
subj.issued	TIMESTAMP	NULLABLE	Publication date.
subj.title	STRING	NULLABLE	The title of the webpage, comment, etc.
subj.author	RECORD	REPEATED	Author of the comment, blog etc.
subj.author.url	STRING	NULLABLE	
subj.author.name	STRING	NULLABLE	
subj.author.id	STRING	NULLABLE	

continues on next page

Table 1 – continued from previous page

name	type	mode	description
subj.url	STRING	NULLABLE	URL where this was found. May be different to 'pid'
subj.alternative_id	STRING	NULLABLE	
subj.original_tweet_author	STRING	NULLABLE	
subj.original_tweet_url	STRING	NULLABLE	
subj.type	STRING	NULLABLE	
subj.work_type_id	STRING	NULLABLE	
subj.work_subtype_id	STRING	NULLABLE	
subj.jurisdiction	STRING	NULLABLE	
subj.api_url	STRING	NULLABLE	
subj.publisher	RECORD	REPEATED	
subj.publisher.url	STRING	NULLABLE	
subj.publisher.name	STRING	NULLABLE	
subj.publisher.id	STRING	NULLABLE	
subj.publisher.type	STRING	NULLABLE	
subj.json_url	STRING	NULLABLE	
subj.name	STRING	NULLABLE	
subj.datePublished	STRING	NULLABLE	
subj.registrantId	STRING	NULLABLE	
subj.dateModified	TIMESTAMP	NULLABLE	
subj.id	STRING	NULLABLE	
subj.proxyIdentifiers	STRING	NULLABLE	
subj.funder	RECORD	NULLABLE	
subj.funder.id	STRING	NULLABLE	
subj.funder.type	STRING	NULLABLE	
subj.funder.name	STRING	NULLABLE	
subj.issueNumber	STRING	NULLABLE	
subj.periodical	RECORD	NULLABLE	
subj.periodical.id	STRING	NULLABLE	
subj.periodical.issn	STRING	NULLABLE	
subj.periodical.type	STRING	NULLABLE	
subj.periodical.name	STRING	NULLABLE	
subj.pagination	STRING	NULLABLE	
subj.version	STRING	NULLABLE	
subj.volumeNumber	STRING	NULLABLE	
subj.includedInDataCatalog	RECORD	NULLABLE	
subj.includedInDataCatalog.id	STRING	NULLABLE	
subj.includedInDataCatalog.type	STRING	NULLABLE	
subj.includedInDataCatalog.name	STRING	NULLABLE	
obj	RECORD	REPEATED	Object metadata.
obj.pid	STRING	NULLABLE	
obj.url	STRING	NULLABLE	
obj.method	STRING	NULLABLE	
obj.verification	STRING	NULLABLE	
obj.work_type_id	STRING	NULLABLE	
obj.publisher	RECORD	REPEATED	
obj.publisher.url	STRING	NULLABLE	
obj.publisher.name	STRING	NULLABLE	

continues on next page

Table 1 – continued from previous page

name	type	mode	description
obj.publisher.id	STRING	NULLABLE	
obj.publisher.type	STRING	NULLABLE	
obj.name	STRING	NULLABLE	
obj.datePublished	STRING	NULLABLE	
obj.registrantId	STRING	NULLABLE	
obj.dateModified	TIMESTAMP	NULLABLE	
obj.id	STRING	NULLABLE	
obj.proxyIdentifiers	STRING	NULLABLE	
obj.author	STRING	NULLABLE	
obj.type	STRING	NULLABLE	
obj.funder	RECORD	NULLABLE	
obj.funder.id	STRING	NULLABLE	
obj.funder.type	STRING	NULLABLE	
obj.funder.name	STRING	NULLABLE	
obj.issueNumber	STRING	NULLABLE	
obj.periodical	RECORD	NULLABLE	
obj.periodical.id	STRING	NULLABLE	
obj.periodical.issn	STRING	NULLABLE	
obj.periodical.type	STRING	NULLABLE	
obj.periodical.name	STRING	NULLABLE	
obj.pagination	STRING	NULLABLE	
obj.version	STRING	NULLABLE	
obj.volumeNumber	STRING	NULLABLE	
obj.includedInDataCatalog	RECORD	NULLABLE	
obj.includedInDataCatalog.id	STRING	NULLABLE	
obj.includedInDataCatalog.type	STRING	NULLABLE	
obj.includedInDataCatalog.name	STRING	NULLABLE	
updated	STRING	NULLABLE	will have a value of 'deleted' or 'edited'
updated_reason	STRING	NULLABLE	optional, may point to an announcement page explaining the edit
updated_date	TIMESTAMP	NULLABLE	ISO8601 date string for when the event was updated
message_action	STRING	NULLABLE	
action	STRING	NULLABLE	
jwt	STRING	NULLABLE	

Crossref metadata

Crossref is a non-for-profit membership organisation working on making scholarly communications better. It is an official Digital Object Identifier (DOI) Registration Agency of the International DOI Foundation. They provide metadata for every DOI that is registered with Crossref.

Crossref Members send Crossref scholarly metadata on research which is collated and standardised into the Crossref metadata dataset. This dataset is made available through services and tools for manuscript tracking, searching, bibliographic management, library systems, author profiling, specialist subject databases, scholarly sharing networks . - *source: Crossref Metadata* and schema details.

The BigQuery table created by the Crossref Metadata telescope from the Academic Observatory workflows is queried with the list of ISBNs from a publisher's Onix feed to create a filtered table in BigQuery called `crossref.crossref_metadataYYYYMMDD`. This table is created during the Onix workflow.

Latest schema

name	type	mode	description
DOI	STRING	NULLABLE	DOI of the work.
ISBN	STRING	REPEATED	
ISSN	STRING	REPEATED	
URL	STRING	NULLABLE	URL form of the work's DOI.
alternative_id	STRING	REPEATED	Other identifiers for the work provided by the depositing member
abstract	STRING	NULLABLE	Abstract as a JSON string or a JATS XML snippet encoded into a JSON string.
author	RECORD	REPEATED	
author.ORCID	STRING	NULLABLE	URL-form of an ORCID identifier
author.affiliation	RECORD	REPEATED	
author.affiliation.acronym	STRING	REPEATED	
author.affiliation.name	STRING	NULLABLE	
author.affiliation.id	RECORD	REPEATED	
author.affiliation.id.id	STRING	NULLABLE	
author.affiliation.id.id_type	STRING	NULLABLE	
author.affiliation.id.asserted _by	STRING	NULLABLE	
author.affiliation.place	STRING	REPEATED	
author.affiliation.department	STRING	REPEATED	
author.authenticated_orcid	BOOLEAN	NULLABLE	If true, record owner asserts that the ORCID user completed ORCID OAuth authentication.
author.family	STRING	NULLABLE	
author.given	STRING	NULLABLE	
author.name	STRING	NULLABLE	
author.sequence	STRING	NULLABLE	
author.suffix	STRING	NULLABLE	
clinical_trial_number	RECORD	REPEATED	
clinical_trial_number.clinical _trial_number	STRING	NULLABLE	Identifier of the clinical trial.
clinical_trial_number.registry	STRING	NULLABLE	DOI of the clinical trial regsitry that assigned the trial number.
clinical_trial_number.type	STRING	NULLABLE	One of preResults, results or postResults
container_title	STRING	REPEATED	Full titles of the containing work (usually a book or journal)
funder	RECORD	REPEATED	
funder.DOI	STRING	NULLABLE	Optional Open Funder Registry DOI uniquely identifing the funding body (http://www.crossref.org/fundingdata/registry.html)
funder.award	STRING	REPEATED	Award number(s) for awards given by the funding body.
funder.doi_asserted_by	STRING	NULLABLE	Either crossref or publisher
funder.name	STRING	NULLABLE	Funding body primary name
group_title	STRING	NULLABLE	Group title for posted content.
is_referenced_by_count	INTEGER	NULLABLE	Count of inbound references deposited with Crossref.
issn_type	RECORD	REPEATED	List of ISSNs with ISSN type information

continues on next page

Table 2 – continued from previous page

name	type	mode	description
issn_type.type	STRING	NULLABLE	ISSN type, can either be print ISSN or electronic ISSN.
issn_type.value	STRING	NULLABLE	ISSN value
issue	STRING	NULLABLE	Issue number of an article's journal.
published_print	RECORD	NULLABLE	
published_print.date_parts	INTEGER	REPEATED	
issued	RECORD	NULLABLE	Earliest of published-print and published-online
issued.date_parts	INTEGER	REPEATED	Contains an ordered array of year, month, day of month. Only year is required. Note that the field contains a nested array, e.g. [[2006, 5, 19]] to conform to citeproc JSON dates
license	RECORD	REPEATED	
license.URL	STRING	NULLABLE	Link to a web page describing this license
license.content_version	STRING	NULLABLE	Either vor (version of record,) am (accepted manuscript,) tdm (text and data mining) or unspecified.
license.delay_in_days	INTEGER	NULLABLE	Number of days between the publication date of the work and the start date of this license.
license.start	RECORD	NULLABLE	Date on which this license begins to take effect
license.start.date_parts	INTEGER	REPEATED	Contains an ordered array of year, month, day of month. Only year is required. Note that the field contains a nested array, e.g. [[2006, 5, 19]] to conform to citeproc JSON dates
license.start.date_time	TIMESTAMP	NULLABLE	ISO 8601 date time.
license.start.timestamp	INTEGER	NULLABLE	Seconds since UNIX epoch.
link	RECORD	REPEATED	URLs to full-text locations.
link.URL	STRING	NULLABLE	Direct link to a full-text download location.
link.content_type	STRING	NULLABLE	Content type (or MIME type) of the full-text object.
link.content_version	STRING	NULLABLE	Either vor (version of record,) am (accepted manuscript) or unspecified.
link.intended_application	STRING	NULLABLE	Either text-mining, similarity-checking or unspecified.
member	INTEGER	NULLABLE	Member identifier of the form http://id.crossref.org/member/MEMBER_ID
page	STRING	NULLABLE	Pages numbers of an article within its journal.
prefix	STRING	NULLABLE	DOI prefix identifier of the form http://id.crossref.org/prefix/DOI_PREFIX.
published	RECORD	NULLABLE	Date on which content was published.
published.date_parts	INTEGER	REPEATED	Contains an ordered array of year, month, day of month. Only year is required. Note that the field contains a nested array, e.g. [[2006, 5, 19]] to conform to citeproc JSON dates

continues on next page

Table 2 – continued from previous page

name	type	mode	description
publisher	STRING	NULLABLE	Name of work's publisher.
publisher_location	STRING	NULLABLE	Location of work's publisher
references_count	INTEGER	NULLABLE	Count of outbound references deposited with Crossref
short_container_title	STRING	REPEATED	Abbreviated titles of the containing work.
subject	STRING	REPEATED	Subject category names, a controlled vocabulary from Sci-Val. Available for most journal articles
title	STRING	REPEATED	Work titles, including translated titles.
type	STRING	NULLABLE	Enumeration, one of the type ids from https://api.crossref.org/v1/types.
volume	STRING	NULLABLE	Volume number of an article's journal.

Google Analytics Universal

Google Analytics was a web analytics service offered by Google that tracks and reports website traffic (now replaced with Google Analytics 4).This telescope gets data from Google Analytics for 1 view id per publisher and for several combinations of metrics and dimensions.It is possible to add a regex expression to filter on pagepaths, so only data on relevant pagepaths is collected.Note that Google Analytics data is only available for the last 26 months, see Data retention - Analytics Help for more info.

To get access to the analytics data a publisher needs to add the relevant google service account as a user.

The corresponding table created in BigQuery is `google.google_analyticsYYYYMMDD`.

Summary	
Average runtime	5 min
Average download size	1 MB
Harvest Type	API
Harvest Frequency	Monthly
Runs on remote worker	False
Catchup missed runs	True
Table Write Disposition	Truncate
Update Frequency	Daily
Credentials Required	Yes
Uses Telescope Template	Snapshot
Each shard includes all data	No

Custom dimensions for ANU Press

ANU Press is using custom dimensions in their google analytics data. To ensure that the telescope processes these custom dimensions, the organisation name needs to be set to exactly 'ANU Press'.The organisation name is used directly inside the telescope and if it matches 'ANU Press' additional dimensions will be added and a different BigQuery schema is used.

A note on the API metrics

We use the python client for the The Google Analytics API in order to retrieve the data on several metrics (such as page views) per country. It appears as though the API does not return a result for every country. We would have expected any data without a country field to be labelled with a country name of **not set**, however this does not appear to be the case.At this time, we have no other way of retrieving country-level data on the desired metrics, so we must acknowledge that the numbers returned by the API are slightly different to those found on the Google Analytics web page.A ticket has been created with google in the hope of resolving this issue.

Telescope object 'extra'

This telescope is created using the Observatory API. There are two 'extra' fields that are required for the corresponding Telescope object.These are the 'view_id' and the 'pagepath_regex'.

view_id

The view_id points to the specific view on which Google Analytics data is collected.See the google support page for more information on the hierarchy of the Analytics account. Below is more information on how to list the view_ids which a service account has access to.

pagepath_regex

This is a regular expression that is used to filter on pagepaths for which analytics data is collected.The regular expression can be set to an empty string if no filtering is required.Note that the Google Analytics API uses 're2', so it is not possible to use e.g. negative lookaheads.See the google support page and github wiki for more information.

Setting up service account

- Create a service account from IAM & Admin - Service Accounts
- Create a JSON key and download the file with key
- For each organisation/publisher of interest, ask them to add this service account as a user for the correct view id

Getting the view ID (after given access)

```
from googleapiclient.discovery import build
from oauth2client.service_account import ServiceAccountCredentials

scopes = ['https://www.googleapis.com/auth/analytics.readonly']
credentials_path = '/path/to/service_account_credentials.json'

creds = ServiceAccountCredentials.from_json_keyfile_name(credentials_path, scopes=scopes)

# Build the service object.
service = build('analytics', 'v3', credentials=creds)

account_summaries = service.management().accountSummaries().list().execute()
view_ids = []
```

(continues on next page)

(continued from previous page)

```
for account in account_summaries['items']:
    account_name = account['name']
    profiles = account['webProperties'][0]['profiles']
    website_url = account['webProperties'][0]['websiteUrl']
    for profile in profiles:
        view_id_info = {'account': account_name, 'websiteUrl': website_url, 'view_id': pr
↪ofile['id'],
                        'view_name': profile['name']}
        view_ids.append(view_id_info)
```

Airflow connections

Note that all values need to be urlencoded. In the config.yaml file, the following airflow connections are required:

oaebu_service_account

After creating the JSON key file as described above, open the JSON file and use the information to create the connection.URL encode each of the fields 'private_key_id', 'private_key', 'client_email' and 'client_id'.

```
oaebu_service_account: google-cloud-platform://?type=service_account&private_key_id=<priv
↪ate_key_id>&private_key=<private_key>&client_email=<client_email>&client_id=<client_id>
```

Latest schema

name	type	mode	description
url	STRING	REQUIRED	Base URL of the book pages.
title	STRING	REQUIRED	Title of the book.
publication_id	STRING	REQUIRED	Custom dimension Publication ID.
publication_type	STRING	REQUIRED	Custom dimension Publication type.
publication_imprint	STRING	REQUIRED	Custom dimension Publication imprint.
publication_group	STRING	REQUIRED	Custom dimension Publication group.
publication_whole_or_part	STRING	REQUIRED	Custom dimension Publication whole/part.
publication_format	STRING	REQUIRED	Custom dimension Publication format.
start_date	DATE	REQUIRED	Start date for period of analytics info.
end_date	DATE	REQUIRED	End date for period of analytics info.
average_time	FLOAT	REQUIRED	Average time (in seconds) spent on each page.
unique_views	RECORD	NULLABLE	Unique views for several different dimensions. Unique views is the number of sessions during which the specified page was viewed at least once. A unique pageview is counted for each page URL + page title combination.
unique_views.country	RECORD	REPEATED	Unique views per users' country, derived from their IP addresses or Geographical IDs.
unique_views.country.name	STRING	NULLABLE	Country name.

continues on next page

Table 3 – continued from previous page

name	type	mode	description
unique_views.country.value	INTEGER	NULLABLE	Number of unique views.
unique_views.referrer	RECORD	REPEATED	Unique views per referrer, the full referring URL including the hostname and path.
unique_views.referrer.name	STRING	NULLABLE	Referrer name.
unique_views.referrer.value	INTEGER	NULLABLE	Number of unique views.
unique_views.social_network	RECORD	REPEATED	Unique views per social network. This is related to the referring social network for traffic sources; e.g., Google+, Blogger.
unique_views.social_network.name	STRING	NULLABLE	Social network name.
unique_views.social_network.value	INTEGER	NULLABLE	Number of unique views.
page_views	RECORD	NULLABLE	The total number of pageviews for the property
page_views.country	RECORD	REPEATED	Page views per users' country, derived from their IP addresses or Geographical IDs.
page_views.country.name	STRING	NULLABLE	Country name.
page_views.country.value	INTEGER	NULLABLE	Number of page views.
page_views.referrer	RECORD	REPEATED	Page views per referrer, the full referring URL including the hostname and path.
page_views.referrer.name	STRING	NULLABLE	Referrer name.
page_views.referrer.value	INTEGER	NULLABLE	Number of page views.
page_views.social_network	RECORD	REPEATED	Page views per social network. This is related to the referring social network for traffic sources; e.g., Google+, Blogger.
page_views.social_network.name	STRING	NULLABLE	Social network name.
page_views.social_network.value	INTEGER	NULLABLE	Number of page views.
sessions	RECORD	NULLABLE	Total number of sessions for several different dimensions.
sessions.country	RECORD	REPEATED	Unique views per users' country, derived from their IP addresses or Geographical IDs.
sessions.country.name	STRING	NULLABLE	Country name.
sessions.country.value	INTEGER	NULLABLE	Number of sessions.
sessions.source	RECORD	REPEATED	Sessions per source of referrals. For manual campaign tracking, it is the value of the utm_source campaign tracking parameter. For AdWords autotagging, it is google. If you use neither, it is the domain of the source (e.g., document.referrer) referring the users. It may also contain a port address. If users arrived without a referrer, its value is (direct)..
sessions.source.name	STRING	NULLABLE	Source name.
sessions.source.value	INTEGER	NULLABLE	Number of sessions.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

Google Books

The Google Books Partner program enables selling books through the Google Play store and offering a preview on Google books.The program makes books discoverable to Google users around the world on Google books. When readers find a book on Google Books, they can preview a limited number of pages to decide if they're interested in it. Readers can also follow links to buy the book or borrow or download it when applicable.

As a publisher you can download reports on Google Books data from https://play.google.com/books/publish/.

Currently there are 3 report types available:

- Google Play sales summary report
- Google Play sales transaction report
- Google Books Traffic Report

In this telescope we collect data from the last 2 reports.The corresponding tables created in BigQuery are `google.google_books_salesYYYYMMDD` and `google.google_books_trafficYYYYMMDD`.

Summary	
Average runtime	5 min
Average download size	1-100 MB
Harvest Type	SFTP
Harvest Frequency	Weekly
Runs on remote worker	False
Catchup missed runs	True
Table Write Disposition	Truncate
Update Frequency	Daily
Credentials Required	Yes
Uses Telescope Template	Snapshot
Each shard includes all data	No

Telescope object 'extra'

This telescope is created using the Observatory API. There is one 'extra' field that is optional for the corresponding Telescope object, namely the 'accounts' field.

accounts

This field is only required if a publisher uses more than 1 Google Books account. If there are multiple accounts for 1 publisher, the reports of these accounts (for the same report type and month) are combined in the 'transform' step of the telescope.To distinguish the reports of the same type and date, but from different accounts, a file suffix is used. When uploading the reports to the SFTP server, this file suffix should be included in the file name. There are instructions both on how to download and correctly name the reports manually as well as how to do it semi -automatically using Selenium.

A list of the file suffixes described above should be passed on to the Telescope 'extra' object.

Authentication

The reports are downloaded from https://play.google.com/books/publish/. To get access to the reports the publisher needs to give access to a google service account.This service account can then be used to login on this webpage and download each report manually.

Setting up a service account

- Create a service account from IAM & Admin - Service Accounts
- Create a JSON key and download the file with key
- For each organisation/publisher of interest, ask them to add this service account for Google Books

Downloading Reports Manually

There is no API available to download the Google Books report and it is quite challenging to automate the Google login process through tools such as Selenium, because of Google's bot detection triggering a reCAPTCHA.Until this step can be automated, the reports need to be downloaded manually.For each publisher and for both the sales transaction report and the traffic report:

- A report should be created for exactly 1 month (e.g. starting 2021-01-01 and ending 2021-01-31).
- All titles should be selected.
- All countries should be selected.
- The traffic report is organised by 'Book'.
- It is important to save the file with the right name, this should be in the following format (<file_suffix> is optional):
  - `GoogleSalesTransactionReport_<file_suffix>YYYY_MM.csv` or
  - `GoogleBooksTrafficReport_<file_suffix>YYYY_MM.csv`
- Upload each report to the SFTP server at https://oaebu.exavault.com/
  - Add it to the folder `/telescopes/google_books/<publisher>/upload`
  - Files are automatically moved between folders, please do not move files between folders manually

Using Selenium to help download reports

When downloading many reports it might be faster to use the script below that helps to download the reports.It is required to run the script in debug mode, so a breakpoint can be set at the right spot (marked in the code) and you can manually login with your Google account.From there on, the reports are automatically downloaded on a monthly basis between the given start and end date, for the given publisher account numbers.To use Selenium you need the chrome webdriver, this can be downloaded from here

```
import os
import shutil
import time

import pendulum
from selenium import webdriver

```

(continues on next page)

(continued from previous page)

```
def main():
    """Download Google Books traffic and sales report using Selenium.
    Needs to be run in debug mode, because it requires manual sign in at breakpoint
↪ (to avoid bot detection).

    Reports are downloaded at a monthly granularity between the start_date and end_date.
    They are downloaded for each publisher in the 'account_numbers' dict and moved t
↪o the corresponding subdirectory
    in the download directory.

    If a publisher has more than 1 account linked a tuple should be used with the p
↪ublisher name and a file suffix.
    The file suffix will be added to the filepath and is used to distinguish report
↪s from different accounts for
    the same publisher.
    The file suffixes that are used here should be passed on to the telescope 'extra
↪' information as described in the
    docs.

    The traffic report is organised by 'Book'.

    :return: None.
    """

    """ Customise values """
    download_dir = "/path/to/download/dir"
    driver_path = "/path/to/chromedriver"
    # Account numbers can be found in the page path when you are signed in to the g
↪oogle books partner center
    account_numbers = {
        "account_number1": "publisher_name1",
        "account_number2": "publisher_name2",
        "account_number3": ("publisher_name3", "suffix1"),
        "account_number4": ("publisher_name3", "suffix2"),
    }
    start_date = pendulum.datetime(2018, 1, 1)
    end_date = pendulum.now()
    """ Customise values """

    # Set download dir for webdriver
    chrome_options = webdriver.ChromeOptions()
    prefs = {"download.default_directory": download_dir}
    chrome_options.add_experimental_option("prefs", prefs)

    # Initialise webdriver and go to books url to login
    driver = webdriver.Chrome(executable_path=driver_path, chrome_options=chrome_options)
    driver.get("https://play.google.com/books/publish/")

    fmt = "%Y,%-m,%-d"  # <-------- set breakpoint here and manually sign in

    # Create download dir
```

(continues on next page)

(continued from previous page)

```
    if not os.path.exists(download_dir):
        os.mkdir(download_dir)

    # Loop through publishers
    for account_number, publisher in account_numbers.items():
        # Get publisher name and file suffix if given
        if isinstance(publisher, tuple):
            name = publisher[0]
            file_suffix = publisher[1]
        else:
            name = publisher
            file_suffix = ""

        # Create publisher dir
        publisher_dir = os.path.join(download_dir, name)
        if not os.path.exists(publisher_dir):
            os.mkdir(publisher_dir)

        # Loop through months
        period = pendulum.period(start_date, end_date)
        for dt in period.range("months"):
            # Skip month if month is not finished yet
            if dt.end_of("month") >= pendulum.now():
                continue

            # Get start and end date in correct string format
            start = dt.strftime(fmt)
            end = dt.end_of("month").strftime(fmt)

            # Download traffic report
            traffic_report_src = os.path.join(download_dir, "GoogleBooksTrafficReport.csv
↪")
            traffic_report_dst = os.path.join(
                publisher_dir, f'GoogleBooksTrafficReport_{file_suffix}{dt.strftime("%Y_%
↪m")}.csv'
            )
            url = (
                f"https://play.google.com/books/publish/u/2/a/{account_number}/downloadTr
↪afficReport?"
                f"f.req=[[null,{start}],[null,{end}],2,false]"
            )
            download_report(driver, url, traffic_report_src, traffic_report_dst)

            # Download sales report
            sales_report_src = os.path.join(download_dir, "GoogleSalesTransactionReport.c
↪sv")
            sales_report_dst = os.path.join(
                publisher_dir,
                f'GoogleSalesTransactionReport_{file_suffix}{dt.strftime("%Y_%m")}.csv',
            )
            url = (
                f"https://play.google.com/books/publish/a/{account_number}/downloadSalesT
```

(continues on next page)

(continued from previous page)

```
↪ransactionReport?"
                f"f.req=[[null,{start}],[null,{end}],[],null,null,null,[],[]]"
            )
            download_report(driver, url, sales_report_src, sales_report_dst)


def download_report(driver: webdriver, url: str, src_path: str, dst_path: str):
    """Download a traffic or sales report from url and move report to a different l
↪ocation.

    :param driver: The chrome webdriver
    :param url: Download url
    :param src_path: File path where file is automatically downloaded to
    :param dst_path: File path where file is moved to
    :return: None.
    """
    # Check if report already exists
    if os.path.exists(dst_path):
        return
    # Download from url
    driver.get(url)
    while not os.path.exists(src_path):
        time.sleep(2)
    # Move to correct dir and add date to filename
    shutil.move(src_path, dst_path)
    print(f"Downloaded: {dst_path}")


if __name__ == "__main__":
    main()
```

Airflow connections

Note that all values need to be urlencoded.In the config.yaml file, the following airflow connection is required:

sftp_service

The sftp_service airflow connection is used to connect to the sftp_service and download the reports.The username and password are created by the sftp service and the host is e.g. `oaebu.exavault.com`.The host key is optional, you can get it by running ssh-keyscan, e.g.:

```
ssh-keyscan oaebu.exavault.com
```

```
sftp_service: ssh://<username>:<password>@<host>:<port>?host_key=<host_key>
```

Latest schema

Google Books Sales

name	type	mode	description
Transaction_Date	DATE	REQUIRED	The date of the transaction.
Id	STRING	REQUIRED	A unique identifier for this transaction.
Product	STRING	NULLABLE	In UCL Press case "Single Purchase" (a normal sale). Can also be "Rental".
Type	STRING	NULLABLE	Type of transaction (can be 'sale' or 'refund').
Preorder	STRING	NULLABLE	Whether this transaction applied to a preorder. In UCL Press case 'None': The transaction didn't involve a preorder.
Qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds.
Primary_ISBN	STRING	NULLABLE	The primary ISBN or other identifier the book, prefixed by a single quotation mark so spreadsheet programs will display the entire ISBN.
Imprint_Name	STRING	REQUIRED	The template used for the book.
Title	STRING	REQUIRED	The title of the book.
Author	STRING	NULLABLE	The author of the book.
Original_List_Price_Currency	STRING	NULLABLE	The original currency of the book's list price.
Original_List_Price	FLOAT	NULLABLE	The original list price of the book.
List_Price_Currency	STRING	NULLABLE	The currency of the book's list price. If currency conversion was enabled, this is the currency of purchase as seen by the buyer.
List_Price_tax_inclusive_	FLOAT	NULLABLE	The book's list prince including tax.
List_Price_tax_exclusive_	FLOAT	NULLABLE	The book's list price excluding tax.
Country_of_Sale	STRING	NULLABLE	The country where the buyer bought the book.
Publisher_Revenue_Perc	FLOAT	NULLABLE	The publisher's percentage of the list price.
Publisher_Revenue	FLOAT	NULLABLE	The amount of revenue earned by the publisher. This will be negative if the transaction was a refund. Negative for refunds. The currency is the same as the payment currency.
Payment_Currency	STRING	NULLABLE	The currency of the publisher's earnings.
Payment_Amount	FLOAT	NULLABLE	The amount earned by the publisher for this transaction. Negative for refunds.
Currency_Conversion_Rate	FLOAT	NULLABLE	If the list price and payment amount are in different currencies, the rate of exchange between the two currencies.
Line_of_Business	STRING	NULLABLE	This field is not present for some publishers (UCL Press). For ANU Press the field value is "E-Book".

continues on next page

Table 4 – continued from previous page

name	type	mode	description
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

Google Books Traffic

name	type	mode	description
Primary_ISBN	STRING	NULLABLE	The primary identifier (e.g., ISBN) of the book. This column appears in the report if data is organized by book.
Title	STRING	REQUIRED	The title of the book.
Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views.
BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books.
Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link).
BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link.
Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages.
Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period). If a user views the same page of your book twice during a session, only a single page view is registered.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

JSTOR

JSTOR provides publisher usage reports, the reports offer details about the use of journal or book content by title, institution, and country. Journal reports also include usage by issue and article. Usage is aligned with the COUNTER 5 standard of Item Requests (views + downloads).Reports can be run or scheduled weekly, monthly, or quarterly with custom date ranges.

To directly get access to the analytics data a publisher needs to grant access to e.g. a Gmail account. This account can then be used to login to the JSTOR portal and set-up the scheduled reports (see below) that are mailed to a G-suite

account.Alternatively, the publisher can set-up a schedule to create reports that are send to the G-suite account.In the telescope the Gmail of the G-suite account is parsed for messages with a download link to the JSTOR report.

The production server of the observatory-platform has been white listed by JSTOR to avoid bot detection.

The corresponding tables created in BigQuery are `jstor.jstor_countryYYYYMMDD` and `jstor.jstor_institutionYYYYMMDD`.

Summary	
Average runtime	5 min
Average download size	5 MB
Harvest Type	API
Harvest Frequency	Monthly
Runs on remote worker	False
Catchup missed runs	True
Table Write Disposition	Truncate
Update Frequency	Daily
Credentials Required	Yes
Uses Telescope Template	Snapshot
Each shard includes all data	No

Telescope object 'extra'

This telescope is created using the Observatory API. There is one 'extra' field that is required for the corresponding Telescope object, namely the 'publisher_id'.

publisher_id

A mapping is required between the JSTOR publisher ID and the organisation name obtained from the observatory API. The JSTOR publisher_id can be found in the original filename of a JSTOR report, for example:`PUB_<publisher_id>_PUBBIU_20210501.tsv`

It is possible to get the original filename by directly downloading a (previous) report from the JSTOR portal.

Setting up a report schedule

Log in to the JSTOR website and set up a report schedule at their portal.It will be easiest to set the report frequency the same as the schedule interval of the telescope. Currently this is set to monthly.For this telescope only the 'Book Usage by Country' (PUB_BCU) and 'Book Usage by Institution' (PUB_BIU) are used.

The format needs to be set to 'TSV' and the recipient to the Gmail account that will be used with the Gmail API.The title of the report is not used in the telescope, so set this to anything you'd like (it does not show up in the email).

Downloading previous reports

Above is described how to set up a report schedule. Unfortunately this schedule can only be set up starting from the current date.To get previous reports (from before the start date of the schedule) it is possible to create a 'one-time' report and mail this to the relevant gmail account. It will then still be processed by this Telescope.The settings are the same as for the scheduled report.

Using Selenium to get previous reports

When downloading many reports it might be faster to use the script below that helps to create the reports.It is required to run the script in debug mode, so a breakpoint can be set at the right spot (marked in the code) and you can manually login with your Google account.From there on, the reports are automatically sent to the gmail account on a monthly basis between the given start and end date, for the given publishers.To use Selenium you need the chrome webdriver, this can be downloaded from here

```
import platform
import time
from datetime import datetime

import pendulum
from selenium import webdriver
from selenium.common.exceptions import ElementClickInterceptedException
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.support.ui import Select


def main():
    """ Create the JSTOR 'Book Usage by Country' and 'Book Usage by Institution' repo
↪rts that are in the past and
    can't be scheduled.
    Needs to be run in debug mode, because it requires manual sign in at breakpoint
↪ (to avoid bot detection).

    Reports are created at a monthly granularity between the start_date and end_date.
    The reports are created for each publisher in the 'publisher_names' list and a l
↪ink to the report is send to the
    email_address.

    There are some 'sleep' statements so the driver waits for the page to load.
    A common issue is that the driver fails at the 'select report' step. The dropdow
↪n menu becomes unclickable. It
    might become clickable after waiting a few seconds or it is necessary to refres
↪h the page and possibly do the
    reCAPTCHA that shows up.

    :return: None.
    """

    """ Customise values """
    driver_path = '/path/to/chromedriver'
    # The publisher name is the exact text displayed when you click 'Select a publisher'
```

(continues on next page)

(continued from previous page)

```
    publisher_names = ["UCL Press (uclpress)", "ANU Press (anuepress)"]
    email_address = 'address@gmail.com'
    start_date = pendulum.datetime(2016, 10, 1)
    end_date = pendulum.now()
    """ Customise values """

    # Initialise webdriver and go to jstor url to login
    driver = webdriver.Chrome(executable_path=driver_path)
    driver.implicitly_wait(10)
    driver.get('https://www.jstor.org/publisher-reports/')

    # Close cookies bar
    try:  # <-------- set breakpoint here and manually sign in
        driver.find_element_by_xpath('//*[@id="onetrust-close-btn-container"]/button').cl
↪ick()
    except:
        pass

    # Click 'create report'
    driver.find_element_by_id('create-report-button').click()

    # Loop through months
    period = pendulum.period(start_date, end_date)
    for dt in period.range('months'):
        # Loop through publishers
        for publisher_name in publisher_names:
            for report_type in ["PUB_BCU", "PUB_BIU"]:
                # Select the publisher
                select_publisher = Select(driver.find_element_by_id('institution-list'))
                select_publisher.select_by_visible_text(publisher_name)

                # Set report type to 'one-time'
                driver.find_element_by_id('is-scheduled-no').click()

                # Select the report type
                time.sleep(5)
                if driver.find_element_by_id('template-list').get_attribute('disabled'):
                    print('pause')  # <----- optionally add breakpoint to wait longer
↪than 5s
                select_report = Select(driver.find_element_by_id('template-list'))
                select_report.select_by_value(report_type)

                # Skip month if month is not finished yet
                if dt.end_of('month') >= pendulum.now():
                    continue

                # Get start and end date of one month
                start_month = dt
                end_month = dt.end_of('month')

                # Set the start and end date
                set_calendar_dates(driver, start_month, end_month)
```

(continues on next page)

(continued from previous page)

```
                # Set the report format
                driver.find_element_by_xpath('//*[@id="available-reports"]/div/fieldset[4
↪]/div[2]/label').click()

                # Fill in email address
                if platform.system() == 'Darwin':
                    key = Keys.COMMAND
                else:
                    key = Keys.CONTROL
                driver.find_element_by_name('email_address').send_keys(key, "a")
                driver.find_element_by_name('email_address').send_keys(email_address)

                # Click continue
                driver.find_element_by_xpath('//*[@id="create-report"]/div[2]/pharos-butt
↪on').click()

                # Click submit, test if duplicate report
                time.sleep(5)
                try:
                    # Submit if not duplicate
                    driver.find_element_by_xpath('//*[@id="create-report"]/div[2]/pharos-
↪button[2]').click()
                    print(f'Created report, name: {publisher_name}, type: {report_type},
↪start: {start_month}, '
                          f'end: {end_month}')
                except ElementClickInterceptedException:
                    # Close if duplicate
                    if driver.find_element_by_xpath('//*[@id="available-reports"]/div/div
↪/div/div['
                                                    '1]/span/strong').text == 'Duplicate
↪report found!':
                        driver.find_element_by_xpath('//*[@id="create-report"]/button').c
↪lick()
                        print(f'Report already exists, name: {publisher_name}, type: {rep
↪ort_type}, '
                              f'start: {start_month}, end: {end_month}')
                    else:
                        raise ElementClickInterceptedException

                # Click 'create report'
                time.sleep(5)
                create_report_button = driver.find_element_by_id('create-report-button')
                action = ActionChains(driver)
                action.move_to_element(create_report_button).click().perform()


def set_calendar_dates(driver: webdriver, start_date: pendulum, end_date: pendulum):
    """ Set the calendar date for the start and end date of the report.

    :param driver: The webdriver
    :param start_date: The start date of this report
```

(continues on next page)

(continued from previous page)

```
    :param end_date: The end dat eof this report
    :return: None.
    """
    date_info = {start_date: {'calendar_id': 'start-calendar',
                              'date_id': 'begin-date',
                              'button': start_date.weekday() + start_date.day},
                 end_date: {'calendar_id': 'end-calendar',
                            'date_id': 'end-date',
                            'button': start_date.weekday() + start_date.day + end_date.da
↪y - 1}}
    for target_date, info in date_info.items():
        calendar_id = info['calendar_id']

        # Click to open calendar
        date_id = driver.find_element_by_id(info['date_id'])
        action = ActionChains(driver)
        action.move_to_element(date_id).click().perform()

        # Find currently set year and month
        set_date_str = driver.find_element_by_xpath(
            f'//*[@id="{calendar_id}"]/div/div[1]/button[3]/span').get_attribute("innerHT
↪ML")
        set_month = datetime.strptime(set_date_str, "%B %Y").month
        set_year = datetime.strptime(set_date_str, "%B %Y").year

        button_map = {'next_year': '5',
                      'previous_year': '1',
                      'next_month': '4',
                      'previous_month': '2'}

        # Go to previous year
        while target_date.year < set_year:
            set_date = go_to(driver, button_map['previous_year'], calendar_id)
            set_year = set_date.year

        # Go to next year
        while target_date.year > set_year:
            set_date = go_to(driver, button_map['next_year'], calendar_id)
            set_year = set_date.year

        # Go to previous month
        while target_date.month < set_month:
            set_date = go_to(driver, button_map['previous_month'], calendar_id)
            set_month = set_date.month

        # Go to next month
        while target_date.month > set_month:
            set_date = go_to(driver, button_map['next_month'], calendar_id)
            set_month = set_date.month

        # Set day, button number starts at
        button = info['button']
```

(continues on next page)

(continued from previous page)

```
        driver.find_element_by_xpath(f'//*[@id="{calendar_id}"]/div/div[2]/div/div/div/di
↪v[2]/button[{button}]').click()

    # Check that set date matches target date
    assert driver.find_element_by_id('begin-date').get_attribute("value") == start_date.s
↪trftime('%Y-%m-%d')
    assert driver.find_element_by_id('end-date').get_attribute("value") == end_date.strft
↪ime('%Y-%m-%d')


def go_to(driver: webdriver, button: str, calendar_id: str) -> datetime:
    """ Click to go to the next/previous month/year

    :param driver: The webdriver
    :param button: The button number corresponding to next or previous and month or year
    :param calendar_id: The calendar id, either 'start calendar' or 'end calendar'
    :return: The month and year that the calendar was set to.
    """
    driver.find_element_by_xpath(f'//*[@id="{calendar_id}"]/div/div[1]/button[{button}]')
↪.click()
    set_date_str = driver.find_element_by_xpath(f'//*[@id="{calendar_id}"]/div/div[1]/but
↪ton[3]/span').text
    set_date = datetime.strptime(set_date_str, "%B %Y")

    return set_date


if __name__ == '__main__':
    main()
```

Using the Gmail API

See the google support answer for info on how to enable an API. Search for the Gmail API and enable this.

Creating the Gmail API connection and credentials

Currently, the telescope works only with a Gmail account that is an internal user (a G-suite account).It is possible to create credentials for an external user with a project status of 'Testing' in the OAuth screen, however refresh tokens created in such a project expire after 7 days and the telescope does not handle expired refresh tokens. See the documentation for more info on OAuth refresh token expiration.

Create OAuth credentials

- In the IAM section add the G-suite account you would like to use as a user.
- From the 'APIs & Services' section, click the 'Credentials' menu item.
- Click 'Create Credentials' and choose OAuth client ID.
- In the form, enter the following information:
  - Application type: Web application
  - Name: Can be anything, e.g. 'Gmail API'
  - Authorized redirect URIs: add the URI: http://localhost:8080/
  - Click 'Create'
- Download the client secrets file for the newly created OAuth 2.0 Client ID, by clicking the download icon for the client ID that you created. The file will be named something like `client_secret_token.apps.googleusercontent.com.json`
- Get the credentials info using the JSON file with client secret info by executing the following python code.

Note that there is currently a limit of 50 refresh tokens per client ID. If the limit is reached, creating a new refresh token automatically invalidates the oldest refresh token without warning. Additionally, tokens are invalidated whenever an account's password is reset.

```
import urllib.parse
from google_auth_oauthlib.flow import InstalledAppFlow

# When modifying these scopes, recreate the file token.json
SCOPES = ['https://www.googleapis.com/auth/gmail.readonly', 'https://www.googleapis.com/a
↪uth/gmail.modify']
flow = InstalledAppFlow.from_client_secrets_file('/path/to/client_secret_token.apps.googl
↪eusercontent.com.json', SCOPES)

# This will open a pop-up, authorize the Gmail account you want to use
creds = flow.run_local_server(access_type='offline', approval_prompt='force', port=8080)

# Get the necessary credentials info
token = urllib.parse.quote(creds.token, safe='')
refresh_token = urllib.parse.quote(creds.refresh_token, safe='')
client_id = urllib.parse.quote(creds.client_id, safe='')
client_secret = urllib.parse.quote(creds.client_secret, safe='')

# This connection can be used in the config file
gmail_api_conn = f'google-cloud-platform://?token={token}&refresh_token={refresh_token}&c
↪lient_id={client_id}&client_secret={client_secret}'
```

Airflow connections

Note that all values need to be urlencoded. In the config.yaml file, the following airflow connections are required:

gmail_api

Use the values from the Gmail API credentials as described above.

```
gmail_api: google-cloud-platform://?token=<token>&refresh_token=<refresh_token>&client_id
↪=<client_id>&client_secret=<client_secret>
```

Latest schema

JSTOR Institution

name	type	mode	description
Institution	STRING	NULLABLE	Institution name.
Book_Title	STRING	REQUIRED	Title of the book.
Book_ID	STRING	NULLABLE	DOI of the book on JSTOR.
Authors	STRING	NULLABLE	Author of the book.
ISBN	STRING	NULLABLE	ISBN of the book (13 digits).
eISBN	STRING	NULLABLE	ISBN of the digital version of the book (13 digits).
Copyright_Year	INTEGER	NULLABLE	Publication year.
Disciplines	STRING	REQUIRED	Subject category of the book.
Usage_Type	STRING	NULLABLE	For our case it is Open Access.
Usage_Month	STRING	REQUIRED	Date (as month and year) of the request.
Total_Item_Requests	INTEGER	NULLABLE	Total number of request made from that specific country.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

JSTOR Country

name	type	mode	description
Country_Name	STRING	NULLABLE	Country Name.
Book_Title	STRING	REQUIRED	Title of the book.
Book_ID	STRING	NULLABLE	DOI of the book on JSTOR.
Authors	STRING	NULLABLE	Author of the book.
ISBN	STRING	NULLABLE	ISBN of the book (13 digits).
eISBN	STRING	NULLABLE	ISBN of the digital version of the book (13 digits).
Copyright_Year	INTEGER	NULLABLE	Publication year.
Disciplines	STRING	REQUIRED	Subject category of the book.
Usage_Type	STRING	NULLABLE	For our case it is Open Access.

continues on next page

Table 7 – continued from previous page

name	type	mode	description
Usage_Month	STRING	REQUIRED	Date (as month and year) of the request.
Total_Item_Requests	INTEGER	NULLABLE	Total number of request made from that specific country.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

OAPEN metadata

The OAPEN Metadata telescope collects data from the OAPEN Metadata feed. OAPEN enables libraries and aggregators to use the metadata of all available titles in the OAPEN Library.The metadata is available in different formats and this telescope harvests the data in the XML format.See the OAPEN Metadata webpage for more information.

The corresponding table in BigQuery is `onix.onixYYYYMMDD`.

Summary	
Average runtime	10min
Average download size	150-200MB
Harvest Type	URL
Harvest Frequency	Weekly
Runs on remote worker	False
Catchup missed runs	False
Table Write Disposition	Append
Update Frequency	Daily
Credentials Required	No
Uses Telescope Template	Stream

Configuration

Airflow Connections

The OAPEN metadata is freely accessible, so no credentials are required for it.

Schedule

The XML file containing metadata is updated daily at +0000GMT. This telescope is scheduled to harvest the metadata weekly.

Results

The resulting ONIX table will be stored in BigQuery - `onix.onixYYYYMMDD`

Tasks

Download

This is where the metadata is downloaded. The XML file containing metadata is downloaded using the XML URL that is available on the OAPEN Metadata webpage mentioned above.

Note that if the metadata file is part-way through an update (occurring daily at +0000GMT and taking upwards of one hour), the XML file will be incomplete and invalid. The telescope has a failesafe to attempt to resolve this during runtime, which can lead to much longer than normal 'download' times.

Transform

The transform step modifies the downloaded metadata into a valid ONIX format. This is done in two steps:

1. The XML is loaded and all unnecessary fields are removed. The fields deemed necessary are described by the header and a supplied product schema (.json file)
2. The resulting XML is parsed through the Python onixcheck. This reveals any remaining invalid products. These products are removed from the file. The removed products are saved to a separate file and uploaded to the transform bucket for storage.

Load to BigQuery

The valid ONIX feed can now be loaded from the transform bucket into a BigQuery sharded table.

Latest schema

name	type	mode	description
CountryOfManufacture	STRING	NULLABLE	An ISO code identifying the country of manufacture of a single-item product, or of a multiple-item product when all items are manufactured in the same country. This information is needed in some countries to meet regulatory requirements. Optional and non-repeating.
RecordSourceName	STRING	NULLABLE	The name of the party which issued the record, as free text. Optional and non-repeating, independently of the occurrence of any other field.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
RecordSourceType	STRING	NULLABLE	An ONIX description which indicates the type of source which has issued the ONIX record. Optional and non- repeating, independently of the occurrence of any other field.
LCCN	STRING	NULLABLE	Library of Congress Control Number
Collections	RECORD	REPEATED	A bibliographic collection in ONIX 3.0 means a fixed or indefinite number of products, published over a fixed or indefinite time period, which share collective attributes (including a collective title) that are required as part of the bibliographic record of each individual product. In this respect, such a collection is most often thought of as a series. A bibliographic collection may, however, also be traded as a single product (often thought of as a set), but this does not alter the way in which its collective attributes are described in the ONIX records for the individual products.
Collections.TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.
Collections.TitleDetails.Title Elements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleElements.PartNumber	RECORD	NULLABLE	When a title element includes a part designation within a larger whole (eg Part I, or Volume 3), this field should be used to carry the number and its 'caption' as text. Optional and non-repeating.
Collections.TitleDetails.TitleElements.PartNumber.Value	STRING	NULLABLE	PartNumber value.
Collections.TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.
Collections.TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.
Collections.TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
Collections.TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> elements is also present.
Collections.TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Collections.TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.
Collections.TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.
Collections.CollectionIdentifers	RECORD	REPEATED	A repeatable group of data elements which together specify an identifier of a bibliographic collection. The composite is optional, and may only repeat if two or more identifiers of different types are sent for the same collection. It is not permissible to have two identifiers of the same type.
Collections.CollectionIdentifers.CollectionIdType	STRING	NULLABLE	An ONIX description identifying a scheme from which an identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.
Collections.CollectionIdentifers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <CollectionIDType> field. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.
Collections.CollectionIdentifers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in <CollectionIDType> indicates a proprietary scheme, eg a publisher's own code. Optional and non-repeating.
Collections.CollectionType	STRING	NULLABLE	An ONIX description indicating the type of a collection: publisher collection, ascribed collection, or unspecified. Mandatory in each occurrence of the <Collection> composite, and non-repeating.
EditionNumber	INTEGER	NULLABLE	The number of a numbered edition. Optional and non-repeating. Normally sent only for the second and subsequent editions of a work, but by agreement between parties to an ONIX exchange a first edition may be explicitly numbered.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
RecordRef	STRING	NULLABLE	Two mandatory data elements must be included at the beginning of every product record or update. The first, <RecordReference>, is a string of text which uniquely identifies the record. The second, <NotificationType>, is a code which specifies the type of notification or update.
RelatedWorks	RECORD	REPEATED	A group of data elements which together describe a work which has a specified relationship to a content item. Optional and repeatable.
RelatedWorks.WorkRelationCode	STRING	NULLABLE	An ONIX description which identifies the nature of the relationship between a product and a work. Mandatory in each occurrence of the <RelatedWork> composite, and non-repeating.
RelatedWorks.WorkIdentifiers	RECORD	REPEATED	A group of data elements which together define an identifier of a work in accordance with a specified scheme. Mandatory in each occurrence of the <RelatedWork> composite, and repeatable only if two or more identifiers for the same work are sent using different identifier schemes (eg ISTC and DOI).
RelatedWorks.WorkIdentifiers.WorkIDType	STRING	NULLABLE	An ONIX description identifying the scheme from which the identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.
RelatedWorks.WorkIdentifiers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <WorkIDType> element. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.
RelatedWorks.WorkIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <WorkIDType> element indicates a proprietary scheme. Optional and non-repeating.
TextContent	RECORD	REPEATED	An optional group of data elements which together carry text related to the product, repeatable in order to deliver multiple texts (often of different types, though for some text types, there may be multiple instances of that type).

continues on next page

Table 8 – continued from previous page

name	type	mode	description
TextContent.TextType	STRING	NULLABLE	An ONIX description which identifies the type of text which is sent in the <Text> element. Mandatory in each occurrence of the <TextContent> composite, and non-repeating.
TextContent.Text	STRING	REPEATED	The text specified in the <TextType> element. Mandatory in each occurrence of the <TextContent> composite, and repeatable when essentially identical text is supplied in multiple languages. The language attribute is optional for a single instance of <Text>, but must be included in each instance if <Text> is repeated.
CityOfPublications	STRING	REPEATED	The name of a city or town associated with the imprint or publisher. Optional, and repeatable if parallel names for a single location appear on the title page in multiple languages, or if the imprint carries two or more cities of publication.
DOI	STRING	NULLABLE	The product's Digital object identifier.
EditionType	STRING	REPEATED	An ONIX description, indicating the type of a version or edition. Optional, and repeatable if the product has characteristics of two or more types (eg 'revised' and 'annotated').
Imprints	RECORD	REPEATED	An optional group of data elements which together identify an imprint or brand under which the product is marketed. The composite must carry either a name identifier or a name or both, and is repeatable to specify multiple imprints or brands.
Imprints.ImprintIdentifiers	RECORD	REPEATED	A group of data elements which together define the identifier of an imprint name. Optional, but mandatory if the <Imprint> composite does not carry an <ImprintName>. The composite is repeatable in order to specify multiple identifiers for the same imprint or brand.
Imprints.ImprintIdentifiers.ImprintIDType	STRING	NULLABLE	An ONIX description which identifies the scheme from which the value in the <IDValue> element is taken. Mandatory in each occurrence of the <ImprintIdentifier> composite.
Imprints.ImprintIdentifiers.IDValue	STRING	NULLABLE	A code value taken from the scheme specified in the <ImprintIDType> element. Mandatory in each occurrence of the <ImprintIdentifier> composite, and non-repeating..

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Imprints.ImprintIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <ImprintIDType> element indicates a proprietary scheme. Optional and non-repeating.
Imprints.ImprintName	STRING	NULLABLE	The name of an imprint or brand under which the product is issued, as it appears on the product. Mandatory if there is no imprint identifier in an occurrence of the <Imprint> composite, and optional if an imprint identifier is included. Non-repeating.
Publishers	RECORD	REPEATED	An optional group of data elements which together identify an entity which is associated with the publishing of a product. The composite allows additional publishing roles to be introduced without adding new fields. Each occurrence of the composite must carry a publishing role code and either a name identifier or a name or both, and the composite is repeatable in order to identify multiple entities.
Publishers.PublisherName	STRING	NULLABLE	The name of an entity associated with the publishing of a product. Mandatory if there is no publisher identifier in an occurrence of the <Publisher> composite, and optional if a publisher identifier is included. Non-repeating.
Publishers.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the publisher identified in an occurrence of the <Publisher> composite. Repeatable in order to provide links to multiple websites.
Publishers.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Publishers.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Publishers.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Publishers.PublishingRole	STRING	NULLABLE	An ONIX description which identifies a role played by an entity in the publishing of a product. Mandatory in each occurrence of the <Publisher> composite, and non-repeating.
RelatedProducts	RECORD	REPEATED	A group of data elements which together describe a product which has a specified relationship to a content item. Optional and repeatable.
RelatedProducts.ISBN13	STRING	NULLABLE	The related product's 13-digit International Standard Book Number.
RelatedProducts.ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
RelatedProducts.DOI	STRING	NULLABLE	The related product's digital object identifier.
RelatedProducts.GTIN_13	STRING	NULLABLE	The related product's 13-digit global trade item number.
RelatedProducts.ProductRelationCodes	STRING	REPEATED	An ONIX description which identifies the nature of the relationship between two products, eg 'replaced-by'. Mandatory in each occurrence of the <RelatedProduct> composite, and repeatable where the related product has multiple types of relationship to the product described.
RelatedProducts.PID_Proprietary	STRING	NULLABLE	The related product's proprietary product ID.
PID_Proprietary	STRING	NULLABLE	The product's proprietary product identifier.
ISBN10	STRING	NULLABLE	The product's 10-digit International Standard Book Number.
ISBN13	STRING	NULLABLE	The product's 13-digit International Standard Book Number.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.
TitleDetails.TitleElements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.
TitleDetails.TitleElements.TitleWithoutPrefix_TextCaseFlags	STRING	NULLABLE	TitleWithoutPrefix textcase attribute.
TitleDetails.TitleElements.TitleText_TextCaseFlags	STRING	NULLABLE	TitleText textcase attribute.
TitleDetails.TitleElements.Subtitle_TextCaseFlags	STRING	NULLABLE	Subtitle textcase attribute.
TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.
TitleDetails.TitleElements.TitleText_Language	STRING	NULLABLE	TitleText language attribute.
TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.
TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> element is also present.
TitleDetails.TitleElements.Subtitle_Language	STRING	NULLABLE	Language attribute.
TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.
TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
TitleDetails.TitleStatement	STRING	NULLABLE	Free text showing how the overall title (including any collection level title, if the collection title is treated as part of the product title and included in P.6) should be presented in any display, particularly when a standard concatenation of individual title elements from Group P.6 (in the order specified by the <SequenceNumber> data elements) would not give a satisfactory result. Optional and non-repeating. When this field is sent, the recipient should use it to replace all title detail sent in Group P.6 for display purposes only. The individual title element detail must also be sent, for indexing and retrieval purposes.
PublishingDates	RECORD	REPEATED	A group of data elements which together specify a date associated with the publishing of the product. Optional, but where known, at least a date of publication must be specified either here (as a 'global' pub date) or in <MarketPublishingDetail> (P.25). Other dates related to the publishing of a product can be sent in further repeats of the composite
PublishingDates.PublishingDateRole	STRING	NULLABLE	An ONIX description indicating the significance of the date, eg publication date, announcement date, latest reprint date. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating.
PublishingDates.Date	INTEGER	NULLABLE	The date specified in the <PublishingDateRole> field. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
GTIN_13	STRING	NULLABLE	The product's 13-digit global trade item number.
Languages	RECORD	REPEATED	A group of data elements which together represent a language, and specify its role and, where required, whether it is a country variant. Optional, and repeatable to specify multiple languages and their various roles.
Languages.CountryCode	STRING	NULLABLE	A code identifying the country when this specifies a variant of the language, eg US English. Optional and non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Languages.LanguageRole	STRING	NULLABLE	An ONIX description indicating the 'role' of a language in the context of the ONIX record. Mandatory in each occurrence of the <Language> composite, and non-repeating.
Languages.LanguageCode	STRING	NULLABLE	An ISO code indicating a language. Mandatory in each occurrence of the <Language> composite, and non-repeating.
ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
Contributors	RECORD	REPEATED	A group of data elements which together describe a personal or corporate contributor to the product. Optional, and repeatable to describe multiple contributors.
Contributors.LettersAfterNames	STRING	NULLABLE	The seventh part of a structured name of a person who contributed to the creation of the product: qualifications and honors following a person's names, eg 'CBE FRS'. Optional and non-repeating.
Contributors.Gender	STRING	NULLABLE	An optional ONIX code specifying the gender of a personal contributor. Not repeatable. Note that this indicates the gender of the contributor's public identity (which may be pseudonymous) based on designations used in ISO 5218, rather than the gender identity, biological sex or sexuality of a natural person.
Contributors.Proprietary	INTEGER	NULLABLE	The contributor's proprietary identifier.
Contributors.NameType	STRING	NULLABLE	An ONIX description indicating the type of a primary name. Optional, and non-repeating. If omitted, the default is 'unspecified'.
Contributors.ProfessionalAffiliations	RECORD	REPEATED	An optional group of data elements which together identify a contributor's professional position and/or affiliation, repeatable to allow multiple positions and affiliations to be specified.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Contributors.ProfessionalAffiliations.Positions	STRING	REPEATED	A professional position held by a contributor to the product at the time of its creation. Optional, and repeatable to provide parallel text in multiple languages. The language attribute is optional for a single instance of <ProfessionalPosition>, but must be included in each instance if <ProfessionalPosition> is repeated.
Contributors.ProfessionalAffiliations.Affiliations	STRING	NULLABLE	An organization to which a contributor to the product was affiliated at the time of its creation, and – if the <ProfessionalPosition> element is also present – where s/he held that position. Optional and non-repeating.
Contributors.ORCID	STRING	NULLABLE	A 16-digit ORCID ID that uniquely identifies the author.
Contributors.BiographicalNotes	RECORD	REPEATED	A biographical note about a contributor to the product. (See the <TextContent> composite in Group P.14 for a biographical note covering all contributors to a product in a single text.) Optional, and repeatable to provide parallel biographical notes in multiple languages. The language attribute is optional for a single instance of <BiographicalNote>, but must be included in each instance if <BiographicalNote> is repeated. May occur with a person name or with a corporate name. A biographical note in ONIX should always contain the name of the person or body concerned, and it should always be presented as a piece of continuous text consisting of full sentences. Some recipients of ONIX data feeds will not accept text which has embedded URLs. A contributor website link can be sent using the <Website> composite below.
Contributors.BiographicalNotes.TextFormat	STRING	NULLABLE	The textformat attribute.
Contributors.BiographicalNotes.Note	STRING	NULLABLE	The biographical note.
Contributors.TitlesBeforeNames	STRING	NULLABLE	The first part of a structured name of a person who contributed to the creation of the product: qualifications and/or titles preceding a person's names, eg 'Professor' or 'HRH Prince' or 'Saint'. Optional and non- repeating: see Group P.7 introductory text for valid options.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Contributors.Roles	STRING	REPEATED	An ONIX description indicating the role played by a person or corporate body in the creation of the product. Mandatory in each occurrence of a <Contributor> composite, and may be repeated if the same person or corporate body has more than one role in relation to the product.
Contributors.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the person or organization identified in an occurrence of the <Contributor> composite. Repeatable to provide links to multiple websites.
Contributors.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Contributors.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.
Contributors.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Contributors.PersonNameInverted	STRING	NULLABLE	The name of a person who contributed to the creation of the product, presented with the element used for alphabetical sorting placed first ('inverted order'). Optional and non-repeating: see Group P.7 introductory text for valid options.
Contributors.Dates	RECORD	REPEATED	A group of data elements which together specify a date associated with the person or organization identified in an occurrence of the <Contributor> composite, eg birth or death. Optional, and repeatable to allow multiple dates to be specified.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Contributors.Dates.Date	INTEGER	NULLABLE	The date specified in the <ContributorDateRole> field. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
Contributors.Dates.Role	STRING	NULLABLE	An ONIX description indicating the significance of the date in relation to the contributor name. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating.
Contributors.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Contributors.PrefixToKey	STRING	NULLABLE	The third part of a structured name of a person who contributed to the creation of the product: a prefix which precedes the key name(s) but which is not to be treated as part of the key name, eg 'van' in Ludwig van Beethoven. This element may also be used for titles that appear after given names and before key names, eg 'Lord' in Alfred, Lord Tennyson. Optional and non-repeating.
Contributors.KeyNames	STRING	NULLABLE	The fourth part of a structured name of a person who contributed to the creation of the product: key name(s), ie the name elements normally used to open an entry in an alphabetical list, eg 'Smith' or 'Garcia Marquez' or 'Madonna' or 'Francis de Sales' (in Saint Francis de Sales). Non-repeating. Required if name part elements P.7.11 to P.7.18 are used.
Contributors.TitlesAfterNames	STRING	NULLABLE	The eighth part of a structured name of a person who contributed to the creation of the product: titles following a person's names, eg 'Duke of Edinburgh'. Optional and non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Contributors.AlternativeNames	STRING	REPEATED	A group of data elements which together represent an alternative name of a contributor, and specify its type. The <AlternativeName> composite is optional, and is repeatable to provide multiple alternative names for the contributor.
Contributors.NamesBeforeKey	STRING	NULLABLE	The second part of a structured name of a person who contributed to the creation of the product: name(s) and/or initial(s) preceding a person's key name(s), eg James J. Optional and non-repeating.
Contributors.Places	RECORD	REPEATED	An optional group of data elements which together identify a geographical location with which a contributor is associated, used to support 'local interest' promotions. Repeatable to identify multiple geographical locations, each usually with a different relationship to the contributor.
Contributors.Places.CountryCode	STRING	NULLABLE	A code identifying a country with which a contributor is particularly associated. Optional and non-repeatable. There must be an occurrence of either the <CountryCode> or the <RegionCode> elements in each occurrence of <ContributorPlace>.
Contributors.Places.Locations	STRING	REPEATED	The name of a city or town location within the specified country or region with which a contributor is particularly associated. Optional, and repeatable to provide parallel names for a single location in multiple languages (eg Baile Átha Cliath and Dublin, or Bruxelles and Brussel). The language attribute is optional for a single instance of <LocationName>, but must be included in each instance if <LocationName> is repeated.
Contributors.Places.Relation	STRING	NULLABLE	An ONIX description identifying the relationship between a contributor and a geographical location. Mandatory in each occurrence of <ContributorPlace> and non-repeating.
Contributors.PersonName	STRING	NULLABLE	The name of a person who contributed to the creation of the product, unstructured, and presented in normal order. Optional and non-repeating: see Group P.7 introductory text for valid options.
Contributors.ISNI	STRING	NULLABLE	16-digit International Standard Name Identifier number.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Contributors.CorprorateName	STRING	NULLABLE	The name of a corporate body which contributed to the creation of the product, unstructured. Optional and non-repeating: see Group P.7 introductory text for valid options.
COKI_ID	STRING	NULLABLE	The product's internal COKI identifier.
Subjects	RECORD	REPEATED	An optional and repeatable group of data elements which together specify a subject classification or subject heading.
Subjects.SubjectHeadingText	STRING	REPEATED	The text of a subject heading taken from the scheme specified in the <SubjectSchemeIdentifier> element, or of free language keywords if the scheme is specified as 'keywords'; or the text equivalent to the <SubjectCode> value, if both code and text are sent. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite.
Subjects.SubjectSchemeIdentifier	STRING	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.
Subjects.SubjectSchemeVersion	FLOAT	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.
Subjects.SubjectSchemeName	STRING	NULLABLE	A name identifying a proprietary subject scheme (ie a scheme which is not a standard and for which there is no individual identifier code) when <SubjectSchemeIdentifier> is coded '24'. Optional and non-repeating.
Subjects.SubjectCode	STRING	NULLABLE	A subject class or category code from the scheme specified in the <SubjectSchemeIdentifier> element. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite. Non-repeating.

continues on next page

Table 8 – continued from previous page

name	type	mode	description
Subjects.MainSubject	BOOLEAN	NULLABLE	An empty element that identifies an instance of the <Subject> composite as representing the main subject category for the product. The main category may be expressed in more than one subject scheme, ie there may be two or more instances of the <Subject> composite, using different schemes, each carrying the <MainSubject/> flag, so long as there is only one main category per scheme. Optional and non-repeating in each occurrence of the <Subject> composite.
Extent	RECORD	REPEATED	A group of data elements which together describe an extent pertaining to the product. Optional, but in practice required for most products, eg to give the number of pages in a printed book or paginated e-book, or to give the running time of an audiobook. Repeatable to specify different extent types or units.
Extent.ExtentType	STRING	NULLABLE	An ONIX description which identifies the type of extent carried in the composite, eg running time for an audio or video product. Mandatory in each occurrence of the <Extent> composite, and non-repeating. From Issue 9 of the code lists, an extended set of values for <ExtentType> has been defined to allow more accurate description of pagination.
Extent.ExtentValue	INTEGER	NULLABLE	The numeric value of the extent specified in <ExtentType>. Optional, and non-repeating. However, either <ExtentValue> or <ExtentValueRoman> must be present in each occurrence of the <Extent> composite; and it is very strongly recommended that <ExtentValue> should always be included, even when the original product uses Roman numerals.
Extent.ExtentUnit	STRING	NULLABLE	An ONIX description indicating the unit used for the <ExtentValue> and the format in which the value is presented. Mandatory in each occurrence of the <Extent> composite, and non-repeating.
Extent.ExtentValueRoman	STRING	NULLABLE	The value of the extent expressed in Roman numerals. Optional, and non-repeating. Used only for page runs which are numbered in Roman.

IRUS OAPEN

IRUS provides OAPEN COUNTER standard access reports. Almost all books on OAPEN are provided as a whole book PDF file. The reports show access figures for each month as well as the location of the access.

Since the location info includes an IP-address, the original data is handled only from within the OAPEN Google Cloud project.

Using a Cloud Function, the original data is downloaded and IP-addresses are replaced with geographical information, such as city and country.After this transformation, the data without IP-addresses is uploaded to a Google Cloud Storage Bucket.

This is all done from within the OAPEN Google Cloud project. The Cloud Function is created and called from the telescope, when the Cloud Function has finished the data is copied from the Storage Bucket inside the OAPEN project, to a Bucket inside the main airflow project.

The corresponding table created in BigQuery is `irus.irus_oapenYYYYMMDD`.

Summary	
Average runtime	5 min
Average download size	5 MB
Harvest Type	API
Harvest Frequency	Monthly
Runs on remote worker	False
Catchup missed runs	True
Table Write Disposition	Truncate
Update Frequency	Daily
Credentials Required	Yes
Uses Telescope Template	Snapshot
Each shard includes all data	No

Telescope object 'extra'

This telescope is created using the Observatory API. There are two 'extra' fields that are required for the corresponding Telescope object, namely the 'publisher_name_v4' and 'publisher_uuid_v5'.A mapping is required between the OAPEN publisher name and the organisation name obtained from the observatory API. The OAPEN publisher name is used directly for the older counter 4 platform, for the newer counter 5 platform the publisher UUID is used.

publisher_name_v4

The publisher_name_v4 can be found by going to the OAPEN page to manually create reports. On this page there is a drop down list with publisher names, to get the publisher name simply url encode the publisher name from this list.

Note that occasionally there are multiple publisher names for one publisher.For example to get all data from Edinburgh University Press, you need data from both publishers `Edinburgh University Press` and `Edinburgh University Press,`. Multiple publisher names can be passed on by delimiting them with a '|' character.

publisher_uuid_v5

The publisher_uuid_v5 can be found by querying the OAPEN API and creating a list of unique Publisher names and UUIDs.

This API request will return all items including their Publisher name and UUID: https://irus.jisc.ac.uk/api/oapen/reports/oapen_ir/?platform=215&requestor_id=<requestor_id>&api_key=<api_key>&granularity=total

To get a file with mappings between Publisher Name and UUID, use the following Python snippet:

```
import requests
import pandas as pd

# Set up your credentials, the start & end date and path to output file
requestor_id = "YOUR_REQUESTOR_ID"
api_key = "YOUR_API_KEY"
start_date = "2020-04"
end_date = "2021-11"
out_file = "/path/to/output_mapping.csv"

# Query the OAPEN API
url = f"https://irus.jisc.ac.uk/api/oapen/reports/oapen_ir/?platform=215&requestor_id={re
↪questor_id}&api_key={api_key}&granularity=totals&begin_date={start_date}&end_date={end_
↪date}"
response = requests.get(url)
response_json = response.json()

# Store result in dataframe, get unique publisher values and sort
df = pd.DataFrame(response_json['Report_Items'])
result = df.drop_duplicates(["Publisher", "Publisher_ID"])[["Publisher", "Publisher_ID"]]
↪.sort_values(["Publisher", "Publisher_ID"])

# Save result to csv file
result.to_csv(out_file, index=False)
```

From this file look up the publisher UUIDs of interest. Similar to the publisher names described above, multiple publisher UUIDs can be passed on by delimiting them with a '|' character.

Cloud Function

The IRUS OAPEN telescope makes use of a Google Cloud Function that resides in the OAPEN Google project.

There is a specific airflow task that will create the Cloud Function if it does not exist yet, or update it if the source code has changed.

The source code for the Cloud Function can be found inside a separate repository that is part of the same organization (https://github.com/The-Academic-Observatory/oapen-irus-uk-cloud-function).

Download access stats data

The Cloud Function downloads IRUS OAPEN access stats data for 1 month and for a single publisher. Usage data after April 2020 is hosted on a new platform.

The newer data is obtained by using their API, this requires a `requestor_id` and an `api_key`.Data before April 2020 is obtained from an URL, this requires an `email` and a `password`.

The required values for either the newer or older way of downloading data are passed on as a `username` and `password` to the Cloud Function. The `username` and `password` are obtained from an airflow connection, which should be set in the config file (see below).

Replace IP addresses

Once the data is downloaded, the IP addresses are replaced with geographical information (corresponding city and country).This is done using the GeoIp database, which is downloaded from inside the Cloud Function. The license key for this database is passed on as a parameter as well, `geoip_license_key`.The `geoip_license_key` is also obtained from an airflow connection, which should be set in the config file (see below).

Upload data to storage bucket

Next, the data without the IP addresses is upload to a bucket inside the OAPEN project. All files in this bucket are deleted after 1 day. In the next airflow task, the data can then be copied from this bucket to the appropriate bucket in the project where airflow is hosted.

Set-up OAPEN Google Cloud project

To make use of the Cloud Function described above it is required to enable two APIs and set up permissions for the Google service account that airflow is using.

See the Google support answer for info on how to enable an API. The API's that need to be enabled are:

- Cloud Functions API
- Cloud build API
- Cloud Run Admin API
- Artifact Registry API

Inside the OAPEN Google project, add the airflow Google service account (<airflow_project_id>@<airflow_project_id>.iam.gserviceaccount.com, where airflow_project_id is the project where airflow is hosted). This can be done from the 'IAM & Admin' menu and 'IAM' tab. Then, assign the following permissions to this account:

- Cloud Functions Developer (to create or update the Cloud Function)
- Cloud Functions Invoker (to call/invoke the Cloud Function)
- Storage Admin (to create a bucket)
- Storage Object Admin (to list and get a blob from the storage bucket)

Additionally, it is required to assign the role of service account user to the service account of the Cloud Function, with the airflow service account as a member. The Cloud SDK command for this is:`gcloud iam service-accounts add-iam-policy-binding <OAPEN_project_id>-compute@developer.gserviceaccount.com`

--member=<airflow_project_id@airflow_project_id.iam.gserviceaccount.com>
--role=roles/iam.serviceAccountUser

Alternatively, it can be done with the Google Cloud console, from the 'IAM & Admin' menu and 'Service Accounts' tab.Click on the service account of the Cloud Function: <OAPEN_project_id>-compute@developer.gserviceaccount.com.In the 'permissions' tab, click 'Grant Access', add the airflow service account as a member <airflow_project_id@airflow_project_id.iam.gserviceaccount.com> and assign the role 'Service Account User'.

Airflow connections

Note that all values need to be urlencoded. In the config.yaml file, the following airflow connections are required:

irus_oapen_login

To get the email address/password combination, contact IRUS.

irus_oapen_api

To get the requestor_id/api_key, contact IRUS.

geoip_license_key

To get the user*id/license_key, first sign up for geolite2 at https://www.maxmind.com/en/geolite2/signup.From your account, in the 'Services' section, click on 'Manage License Keys'. The user_id is displayed on this page.Then, click on 'Generate new license key', this can be used for the 'license_key'.Answer _No* for the question: "Old versions of our GeoIP Update program use a different license key format. Will this key be used for GeoIP Update?"

```
irus_oapen_login: mysql://email_address:password@
irus_oapen_api: mysql://requestor_id:api_key@
geoip_license_key: mysql://user_id:license_key@
```

Latest schema

name	type	mode	description
proprietary_id	STRING	NULLABLE	Proprietary identifier of the book.
URI	STRING	NULLABLE	URI of the book. Only available for data since 2020-04-01.
DOI	STRING	NULLABLE	DOI of the book.
ISBN	STRING	NULLABLE	ISBN of the book.
book_title	STRING	NULLABLE	Title of the book
grant	STRING	NULLABLE	Grant. Only available for data before 2020-04-01.
grant_number	STRING	NULLABLE	Grant number. Only available for data before 2020-04-01.
publisher	STRING	NULLABLE	The publisher

continues on next page

Table 9 – continued from previous page

name	type	mode	description
begin_date	DATE	NULLABLE	The begin date of the investigated period.
end_date	DATE	NULLABLE	The end date of the investigated period.
title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01.
total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01.
total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01.
unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01.
unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01.
country	RECORD	REPEATED	Record to store statistics on the country level.
country.name	STRING	NULLABLE	The country name of the client registered by oapen irus uk.
country.code	STRING	NULLABLE	The country code of the client registered by oapen irus uk.
country.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01.
country.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01.
country.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01.
country.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01.
country.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01.
locations	RECORD	REPEATED	Record to store statistics on the location level.
locations.latitude	FLOAT	NULLABLE	The latitude geolocated from the client's ip address.
locations.longitude	FLOAT	NULLABLE	The longitude geolocated from the client's ip address.
locations.city	STRING	NULLABLE	The city geolocated from the client's ip address.
locations.country_name	STRING	NULLABLE	The country name geolocated from the client's ip address.
locations.country_code	STRING	NULLABLE	The country code geolocated from the client's ip address.
locations.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01.
locations.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01.
locations.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01.
locations.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01.
locations.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01.

continues on next page

Table 9 – continued from previous page

name	type	mode	description
version	STRING	REQUIRED	Version of the OAPEN IRUS UK API, corresponds to the COUNTER report version.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

IRUS Fulcrum

The IRUS Fulcrum telescope collects usage statistics for titles accessed via the Fulcrum Platform. Usage data is accessible through IRUS in much the same way as the IRUS OAPEN telescope. Unlike IRUS OAPEN, IRUS Fulcrum does not record sensitive IP address information. This makes dealing with the data much simpler.

The earliest available data for the Fulcrum platform is April 2022. It follows that all data is of COUNTER 5 standard.

The corresponding table created in BigQuery is `irus.irus_fulcrumYYYYMMDD`.

Summary	
Average runtime	5-10 mins
Average download size	1-10 MB
Harvest Type	API
Harvest Frequency	Monthly
Runs on remote worker	False
Catchup missed runs	True
Credentials Required	Yes
Uses Telescope Template	None
Each shard includes all data	No

Airflow connections

Note that all values need to be urlencoded. In the config.yaml file, the following airflow connections are required:

Irus_api

The IRUS requestor_id/api_key is required to access the IRUS platform.

Data Download

The download is done via an API call to IRUS:

```
https://irus.jisc.ac.uk/api/v3/irus/reports/irus_ir/?platform=235&requestor_id={requestor
↪_id}&begin_date={start_date}&end_date={end_date}
```

Where the requestor ID is the API key for the IRUS API. The telescope will use the same begin and end dates (YYYY-MM) in order to retrieve data on a per-month basis.

A second call to the API is made with the following appended to the above URL:

```
&attributes_to_show=Country
```

Which splits the data by country, leaving us with two datasets. These datasets will be referred to as the *total* and *country* datasets.

Before making any changes to the data, these datasets are uploaded to a Google storage bucket

Data Transform

The transform step has a few things to achieve:

- Collate the *total* and *country* datasets into a single object
- Remove columns that are not of interest to us
- Add the release month to each row as a partitioning column
- Remove rows from the data that do not relate to the publisher of interest

The result of points 1 -> 3 are evident in the *schema*. The final point requires some communication with the publisher. This is because a single publisher may have published titles under more than one name. For example, University of Michigan has 10 associated publishing names. These names are listed as part of a dictionary in the telescope.

The resulting transformed file is uploaded to a Google Cloud bucket

BigQuery Load

The transformed data is loaded from the Google Cloud bucket into a partitioned BigQuery table. The table is in the respective publisher's Project and a *fulcrum* dataset will be created if it does not exist. Since the data is partitioned on the release month, there will only be a single table.

Latest schema

name	type	mode	description
proprietary_id	STRING	NULLABLE	Proprietary identifier of the book.
ISBN	STRING	NULLABLE	ISBN of the book.
book_title	STRING	NULLABLE	Title of the book
publisher	STRING	NULLABLE	The publisher
authors	STRING	NULLABLE	The names of the authors
event_month	STRING	NULLABLE	The investigated month.
total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
total_item_requests	INTEGER	NULLABLE	The total number of item requests.
unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.
country	RECORD	REPEATED	Record to store statistics on the country level.
country.name	STRING	NULLABLE	The country name of the client registered by IRUS.
country.code	STRING	NULLABLE	The country code of the client registered by IRUS.
country.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.

continues on next page

Table 10 – continued from previous page

name	type	mode	description
country.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
country.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
country.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

ONIX

The ONIX telescope downloads, transforms and loads publisher ONIX feeds into BigQuery. ONIX is a standard format that book publishers use to share information about the books that they have published.

Book publishers with ONIX feeds are given credentials and access to their own upload folder on the OAeBU SFTP server. They then configure ONIX Suite to upload their ONIX feeds to the SFTP server on a weekly basis. The ONIX feeds need to be full dumps every time, not incremental updates.

The ONIX telescope downloads the ONIX files from the SFTP server. It then transforms the data into a format suitable for loading into BigQuery with the ONIX parser Java command line tool. The data is loaded into BigQuery and then used by the ONIX Workflow.

The corresponding table in BigQuery is `onix.onixYYYYMMDD`.

Summary	
Average runtime	10-20 mins
Average download size	10-100 MB
Harvest Type	SFTP server
Harvest Frequency	Weekly
Runs on remote worker	False
Catchup missed runs	False
Credentials Required	Yes
Uses Telescope Template	Snapshot
Each shard includes all data	Yes

Configuration

The following settings need to be configured for the ONIX telescope.

Telescope API instance

An ONIX telescope API instance needs to be created, making sure to add the below settings to the extra field.

Telescope API 'extra' field

The `date_regex` field must be added to the telescope extra field, it is used to extract the date from the ONIX feed file name. For example, the regex `\\d{8}` will extract the date from the file name `20220301_CURTINPRESS_ONIX.xml`.

```
"extra": {
  "date_regex": "\\d{8}"
}
```

Airflow connections

In the config.yaml file, the below Airflow connection is required. Note that all values need to be urlencoded.

sftp_service

The ssh username, password and host key to connect to the SFTP server.

```
sftp_service: ssh://user-name:password@host-name:port?host_key=host-key
```

Latest schema

name	type	mode	description
CountryOfManufacture	STRING	NULLABLE	An ISO code identifying the country of manufacture of a single-item product, or of a multiple-item product when all items are manufactured in the same country. This information is needed in some countries to meet regulatory requirements. Optional and non-repeating.
RecordSourceName	STRING	NULLABLE	The name of the party which issued the record, as free text. Optional and non-repeating, independently of the occurrence of any other field.
RecordSourceType	STRING	NULLABLE	An ONIX description which indicates the type of source which has issued the ONIX record. Optional and non- repeating, independently of the occurrence of any other field.
LCCN	STRING	NULLABLE	Library of Congress Control Number

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Collections	RECORD	REPEATED	A bibliographic collection in ONIX 3.0 means a fixed or indefinite number of products, published over a fixed or indefinite time period, which share collective attributes (including a collective title) that are required as part of the bibliographic record of each individual product. In this respect, such a collection is most often thought of as a series. A bibliographic collection may, however, also be traded as a single product (often thought of as a set), but this does not alter the way in which its collective attributes are described in the ONIX records for the individual products.
Collections.TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.
Collections.TitleDetails.Title Elements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.
Collections.TitleDetails.Title Elements.PartNumber	RECORD	NULLABLE	When a title element includes a part designation within a larger whole (eg Part I, or Volume 3), this field should be used to carry the number and its 'caption' as text. Optional and non-repeating.
Collections.TitleDetails.Title Elements.PartNumber.Value	STRING	NULLABLE	PartNumber value.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.
Collections.TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.
Collections.TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
Collections.TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> elements is also present.
Collections.TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Collections.TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.
Collections.TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.
Collections.CollectionIdentifers	RECORD	REPEATED	A repeatable group of data elements which together specify an identifier of a bibliographic collection. The composite is optional, and may only repeat if two or more identifiers of different types are sent for the same collection. It is not permissible to have two identifiers of the same type.
Collections.CollectionIdentifers.CollectionIdType	STRING	NULLABLE	An ONIX description identifying a scheme from which an identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.
Collections.CollectionIdentifers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <CollectionIDType> field. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.
Collections.CollectionIdentifers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in <CollectionIDType> indicates a proprietary scheme, eg a publisher's own code. Optional and non-repeating.
Collections.CollectionType	STRING	NULLABLE	An ONIX description indicating the type of a collection: publisher collection, ascribed collection, or unspecified. Mandatory in each occurrence of the <Collection> composite, and non-repeating.
EditionNumber	INTEGER	NULLABLE	The number of a numbered edition. Optional and non-repeating. Normally sent only for the second and subsequent editions of a work, but by agreement between parties to an ONIX exchange a first edition may be explicitly numbered.
RecordRef	STRING	NULLABLE	Two mandatory data elements must be included at the beginning of every product record or update. The first, <RecordReference>, is a string of text which uniquely identifies the record. The second, <NotificationType>, is a code which specifies the type of notification or update.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
RelatedWorks	RECORD	REPEATED	A group of data elements which together describe a work which has a specified relationship to a content item. Optional and repeatable.
RelatedWorks.WorkRelationCode	STRING	NULLABLE	An ONIX description which identifies the nature of the relationship between a product and a work. Mandatory in each occurrence of the <RelatedWork> composite, and non-repeating.
RelatedWorks.WorkIdentifiers	RECORD	REPEATED	A group of data elements which together define an identifier of a work in accordance with a specified scheme. Mandatory in each occurrence of the <RelatedWork> composite, and repeatable only if two or more identifiers for the same work are sent using different identifier schemes (eg ISTC and DOI).
RelatedWorks.WorkIdentifiers.WorkIDType	STRING	NULLABLE	An ONIX description identifying the scheme from which the identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.
RelatedWorks.WorkIdentifiers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <WorkIDType> element. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.
RelatedWorks.WorkIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <WorkIDType> element indicates a proprietary scheme. Optional and non-repeating.
TextContent	RECORD	REPEATED	An optional group of data elements which together carry text related to the product, repeatable in order to deliver multiple texts (often of different types, though for some text types, there may be multiple instances of that type).
TextContent.TextType	STRING	NULLABLE	An ONIX description which identifies the type of text which is sent in the <Text> element. Mandatory in each occurrence of the <TextContent> composite, and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
TextContent.Text	STRING	REPEATED	The text specified in the <TextType> element. Mandatory in each occurrence of the <TextContent> composite, and repeatable when essentially identical text is supplied in multiple languages. The language attribute is optional for a single instance of <Text>, but must be included in each instance if <Text> is repeated.
CityOfPublications	STRING	REPEATED	The name of a city or town associated with the imprint or publisher. Optional, and repeatable if parallel names for a single location appear on the title page in multiple languages, or if the imprint carries two or more cities of publication.
DOI	STRING	NULLABLE	The product's Digital object identifier.
EditionType	STRING	REPEATED	An ONIX description, indicating the type of a version or edition. Optional, and repeatable if the product has characteristics of two or more types (eg 'revised' and 'annotated').
Imprints	RECORD	REPEATED	An optional group of data elements which together identify an imprint or brand under which the product is marketed. The composite must carry either a name identifier or a name or both, and is repeatable to specify multiple imprints or brands.
Imprints.ImprintIdentifiers	RECORD	REPEATED	A group of data elements which together define the identifier of an imprint name. Optional, but mandatory if the <Imprint> composite does not carry an <ImprintName>. The composite is repeatable in order to specify multiple identifiers for the same imprint or brand.
Imprints.ImprintIdentifiers.ImprintIDType	STRING	NULLABLE	An ONIX description which identifies the scheme from which the value in the <IDValue> element is taken. Mandatory in each occurrence of the <ImprintIdentifier> composite.
Imprints.ImprintIdentifiers.IDValue	STRING	NULLABLE	A code value taken from the scheme specified in the <ImprintIDType> element. Mandatory in each occurrence of the <ImprintIdentifier> composite, and non-repeating..

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Imprints.ImprintIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <ImprintIDType> element indicates a proprietary scheme. Optional and non-repeating.
Imprints.ImprintName	STRING	NULLABLE	The name of an imprint or brand under which the product is issued, as it appears on the product. Mandatory if there is no imprint identifier in an occurrence of the <Imprint> composite, and optional if an imprint identifier is included. Non-repeating.
Publishers	RECORD	REPEATED	An optional group of data elements which together identify an entity which is associated with the publishing of a product. The composite allows additional publishing roles to be introduced without adding new fields. Each occurrence of the composite must carry a publishing role code and either a name identifier or a name or both, and the composite is repeatable in order to identify multiple entities.
Publishers.PublisherName	STRING	NULLABLE	The name of an entity associated with the publishing of a product. Mandatory if there is no publisher identifier in an occurrence of the <Publisher> composite, and optional if a publisher identifier is included. Non-repeating.
Publishers.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the publisher identified in an occurrence of the <Publisher> composite. Repeatable in order to provide links to multiple websites.
Publishers.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Publishers.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Publishers.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Publishers.PublishingRole	STRING	NULLABLE	An ONIX description which identifies a role played by an entity in the publishing of a product. Mandatory in each occurrence of the <Publisher> composite, and non-repeating.
RelatedProducts	RECORD	REPEATED	A group of data elements which together describe a product which has a specified relationship to a content item. Optional and repeatable.
RelatedProducts.ISBN13	STRING	NULLABLE	The related product's 13-digit International Standard Book Number.
RelatedProducts.ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
RelatedProducts.DOI	STRING	NULLABLE	The related product's digital object identifier.
RelatedProducts.GTIN_13	STRING	NULLABLE	The related product's 13-digit global trade item number.
RelatedProducts.ProductRelationCodes	STRING	REPEATED	An ONIX description which identifies the nature of the relationship between two products, eg 'replaced-by'. Mandatory in each occurrence of the <RelatedProduct> composite, and repeatable where the related product has multiple types of relationship to the product described.
RelatedProducts.PID_Proprietary	STRING	NULLABLE	The related product's proprietary product ID.
PID_Proprietary	STRING	NULLABLE	The product's proprietary product identifier.
ISBN10	STRING	NULLABLE	The product's 10-digit International Standard Book Number.
ISBN13	STRING	NULLABLE	The product's 13-digit International Standard Book Number.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.
TitleDetails.TitleElements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.
TitleDetails.TitleElements.TitleWithoutPrefix_TextCaseFlags	STRING	NULLABLE	TitleWithoutPrefix textcase attribute.
TitleDetails.TitleElements.TitleText_TextCaseFlags	STRING	NULLABLE	TitleText textcase attribute.
TitleDetails.TitleElements.Subtitle_TextCaseFlags	STRING	NULLABLE	Subtitle textcase attribute.
TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.
TitleDetails.TitleElements.TitleText_Language	STRING	NULLABLE	TitleText language attribute.
TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.
TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> element is also present.
TitleDetails.TitleElements.Subtitle_Language	STRING	NULLABLE	Language attribute.
TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.
TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
TitleDetails.TitleStatement	STRING	NULLABLE	Free text showing how the overall title (including any collection level title, if the collection title is treated as part of the product title and included in P.6) should be presented in any display, particularly when a standard concatenation of individual title elements from Group P.6 (in the order specified by the <SequenceNumber> data elements) would not give a satisfactory result. Optional and non-repeating. When this field is sent, the recipient should use it to replace all title detail sent in Group P.6 for display purposes only. The individual title element detail must also be sent, for indexing and retrieval purposes.
PublishingDates	RECORD	REPEATED	A group of data elements which together specify a date associated with the publishing of the product. Optional, but where known, at least a date of publication must be specified either here (as a 'global' pub date) or in <MarketPublishingDetail> (P.25). Other dates related to the publishing of a product can be sent in further repeats of the composite
PublishingDates.PublishingDateRole	STRING	NULLABLE	An ONIX description indicating the significance of the date, eg publication date, announcement date, latest reprint date. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating.
PublishingDates.Date	INTEGER	NULLABLE	The date specified in the <PublishingDateRole> field. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
GTIN_13	STRING	NULLABLE	The product's 13-digit global trade item number.
Languages	RECORD	REPEATED	A group of data elements which together represent a language, and specify its role and, where required, whether it is a country variant. Optional, and repeatable to specify multiple languages and their various roles.
Languages.CountryCode	STRING	NULLABLE	A code identifying the country when this specifies a variant of the language, eg US English. Optional and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Languages.LanguageRole	STRING	NULLABLE	An ONIX description indicating the 'role' of a language in the context of the ONIX record. Mandatory in each occurrence of the <Language> composite, and non-repeating.
Languages.LanguageCode	STRING	NULLABLE	An ISO code indicating a language. Mandatory in each occurrence of the <Language> composite, and non-repeating.
ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
Contributors	RECORD	REPEATED	A group of data elements which together describe a personal or corporate contributor to the product. Optional, and repeatable to describe multiple contributors.
Contributors.LettersAfterNames	STRING	NULLABLE	The seventh part of a structured name of a person who contributed to the creation of the product: qualifications and honors following a person's names, eg 'CBE FRS'. Optional and non-repeating.
Contributors.Gender	STRING	NULLABLE	An optional ONIX code specifying the gender of a personal contributor. Not repeatable. Note that this indicates the gender of the contributor's public identity (which may be pseudonymous) based on designations used in ISO 5218, rather than the gender identity, biological sex or sexuality of a natural person.
Contributors.Proprietary	INTEGER	NULLABLE	The contributor's proprietary identifier.
Contributors.NameType	STRING	NULLABLE	An ONIX description indicating the type of a primary name. Optional, and non-repeating. If omitted, the default is 'unspecified'.
Contributors.ProfessionalAffiliations	RECORD	REPEATED	An optional group of data elements which together identify a contributor's professional position and/or affiliation, repeatable to allow multiple positions and affiliations to be specified.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Contributors.ProfessionalAffiliations.Positions	STRING	REPEATED	A professional position held by a contributor to the product at the time of its creation. Optional, and repeatable to provide parallel text in multiple languages. The language attribute is optional for a single instance of <ProfessionalPosition>, but must be included in each instance if <ProfessionalPosition> is repeated.
Contributors.ProfessionalAffiliations.Affiliations	STRING	NULLABLE	An organization to which a contributor to the product was affiliated at the time of its creation, and – if the <ProfessionalPosition> element is also present – where s/he held that position. Optional and non-repeating.
Contributors.ORCID	STRING	NULLABLE	A 16-digit ORCID ID that uniquely identifies the author.
Contributors.BiographicalNotes	RECORD	REPEATED	A biographical note about a contributor to the product. (See the <TextContent> composite in Group P.14 for a biographical note covering all contributors to a product in a single text.) Optional, and repeatable to provide parallel biographical notes in multiple languages. The language attribute is optional for a single instance of <BiographicalNote>, but must be included in each instance if <BiographicalNote> is repeated. May occur with a person name or with a corporate name. A biographical note in ONIX should always contain the name of the person or body concerned, and it should always be presented as a piece of continuous text consisting of full sentences. Some recipients of ONIX data feeds will not accept text which has embedded URLs. A contributor website link can be sent using the <Website> composite below.
Contributors.BiographicalNotes.TextFormat	STRING	NULLABLE	The textformat attribute.
Contributors.BiographicalNotes.Note	STRING	NULLABLE	The biographical note.
Contributors.TitlesBeforeNames	STRING	NULLABLE	The first part of a structured name of a person who contributed to the creation of the product: qualifications and/or titles preceding a person's names, eg 'Professor' or 'HRH Prince' or 'Saint'. Optional and non- repeating: see Group P.7 introductory text for valid options.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Contributors.Roles	STRING	REPEATED	An ONIX description indicating the role played by a person or corporate body in the creation of the product. Mandatory in each occurrence of a <Contributor> composite, and may be repeated if the same person or corporate body has more than one role in relation to the product.
Contributors.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the person or organization identified in an occurrence of the <Contributor> composite. Repeatable to provide links to multiple websites.
Contributors.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Contributors.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.
Contributors.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Contributors.PersonNameInverted	STRING	NULLABLE	The name of a person who contributed to the creation of the product, presented with the element used for alphabetical sorting placed first ('inverted order'). Optional and non-repeating: see Group P.7 introductory text for valid options.
Contributors.Dates	RECORD	REPEATED	A group of data elements which together specify a date associated with the person or organization identified in an occurrence of the <Contributor> composite, eg birth or death. Optional, and repeatable to allow multiple dates to be specified.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Contributors.Dates.Date	INTEGER	NULLABLE	The date specified in the <ContributorDateRole> field. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
Contributors.Dates.Role	STRING	NULLABLE	An ONIX description indicating the significance of the date in relation to the contributor name. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating.
Contributors.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Contributors.PrefixToKey	STRING	NULLABLE	The third part of a structured name of a person who contributed to the creation of the product: a prefix which precedes the key name(s) but which is not to be treated as part of the key name, eg 'van' in Ludwig van Beethoven. This element may also be used for titles that appear after given names and before key names, eg 'Lord' in Alfred, Lord Tennyson. Optional and non-repeating.
Contributors.KeyNames	STRING	NULLABLE	The fourth part of a structured name of a person who contributed to the creation of the product: key name(s), ie the name elements normally used to open an entry in an alphabetical list, eg 'Smith' or 'Garcia Marquez' or 'Madonna' or 'Francis de Sales' (in Saint Francis de Sales). Non-repeating. Required if name part elements P.7.11 to P.7.18 are used.
Contributors.TitlesAfterNames	STRING	NULLABLE	The eighth part of a structured name of a person who contributed to the creation of the product: titles following a person's names, eg 'Duke of Edinburgh'. Optional and non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Contributors.AlternativeNames	STRING	REPEATED	A group of data elements which together represent an alternative name of a contributor, and specify its type. The <AlternativeName> composite is optional, and is repeatable to provide multiple alternative names for the contributor.
Contributors.NamesBeforeKey	STRING	NULLABLE	The second part of a structured name of a person who contributed to the creation of the product: name(s) and/or initial(s) preceding a person's key name(s), eg James J. Optional and non-repeating.
Contributors.Places	RECORD	REPEATED	An optional group of data elements which together identify a geographical location with which a contributor is associated, used to support 'local interest' promotions. Repeatable to identify multiple geographical locations, each usually with a different relationship to the contributor.
Contributors.Places.CountryCode	STRING	NULLABLE	A code identifying a country with which a contributor is particularly associated. Optional and non-repeatable. There must be an occurrence of either the <CountryCode> or the <RegionCode> elements in each occurrence of <ContributorPlace>.
Contributors.Places.Locations	STRING	REPEATED	The name of a city or town location within the specified country or region with which a contributor is particularly associated. Optional, and repeatable to provide parallel names for a single location in multiple languages (eg Baile Átha Cliath and Dublin, or Bruxelles and Brussel). The language attribute is optional for a single instance of <LocationName>, but must be included in each instance if <LocationName> is repeated.
Contributors.Places.Relation	STRING	NULLABLE	An ONIX description identifying the relationship between a contributor and a geographical location. Mandatory in each occurrence of <ContributorPlace> and non-repeating.
Contributors.PersonName	STRING	NULLABLE	The name of a person who contributed to the creation of the product, unstructured, and presented in normal order. Optional and non-repeating: see Group P.7 introductory text for valid options.
Contributors.ISNI	STRING	NULLABLE	16-digit International Standard Name Identifier number.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Contributors.CorprorateName	STRING	NULLABLE	The name of a corporate body which contributed to the creation of the product, unstructured. Optional and non-repeating: see Group P.7 introductory text for valid options.
COKI_ID	STRING	NULLABLE	The product's internal COKI identifier.
Subjects	RECORD	REPEATED	An optional and repeatable group of data elements which together specify a subject classification or subject heading.
Subjects.SubjectHeadingText	STRING	REPEATED	The text of a subject heading taken from the scheme specified in the <SubjectSchemeIdentifier> element, or of free language keywords if the scheme is specified as 'keywords'; or the text equivalent to the <SubjectCode> value, if both code and text are sent. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite.
Subjects.SubjectSchemeIdentifier	STRING	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.
Subjects.SubjectSchemeVersion	FLOAT	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.
Subjects.SubjectSchemeName	STRING	NULLABLE	A name identifying a proprietary subject scheme (ie a scheme which is not a standard and for which there is no individual identifier code) when <SubjectSchemeIdentifier> is coded '24'. Optional and non-repeating.
Subjects.SubjectCode	STRING	NULLABLE	A subject class or category code from the scheme specified in the <SubjectSchemeIdentifier> element. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite. Non-repeating.

continues on next page

Table 11 – continued from previous page

name	type	mode	description
Subjects.MainSubject	BOOLEAN	NULLABLE	An empty element that identifies an instance of the <Subject> composite as representing the main subject category for the product. The main category may be expressed in more than one subject scheme, ie there may be two or more instances of the <Subject> composite, using different schemes, each carrying the <MainSubject/> flag, so long as there is only one main category per scheme. Optional and non-repeating in each occurrence of the <Subject> composite.
Extent	RECORD	REPEATED	A group of data elements which together describe an extent pertaining to the product. Optional, but in practice required for most products, eg to give the number of pages in a printed book or paginated e-book, or to give the running time of an audiobook. Repeatable to specify different extent types or units.
Extent.ExtentType	STRING	NULLABLE	An ONIX description which identifies the type of extent carried in the composite, eg running time for an audio or video product. Mandatory in each occurrence of the <Extent> composite, and non-repeating. From Issue 9 of the code lists, an extended set of values for <ExtentType> has been defined to allow more accurate description of pagination.
Extent.ExtentValue	INTEGER	NULLABLE	The numeric value of the extent specified in <ExtentType>. Optional, and non-repeating. However, either <ExtentValue> or <ExtentValueRoman> must be present in each occurrence of the <Extent> composite; and it is very strongly recommended that <ExtentValue> should always be included, even when the original product uses Roman numerals.
Extent.ExtentUnit	STRING	NULLABLE	An ONIX description indicating the unit used for the <ExtentValue> and the format in which the value is presented. Mandatory in each occurrence of the <Extent> composite, and non-repeating.
Extent.ExtentValueRoman	STRING	NULLABLE	The value of the extent expressed in Roman numerals. Optional, and non-repeating. Used only for page runs which are numbered in Roman.

Thoth

The Thoth Telescope downloads, transforms and loads publisher ONIX feeds from Thoth into BigQuery. ONIX is a standard format that book publishers use to share information about the books that they have published.

Thoth is a free, open metadata service that publishers can choose to utilise as a solution for metadata storage. Thoth can provide metadata upon request in a number of formats. The Thoth Telescope used the Thoth Export API to download metadata in an ONIX format. This API provides a snapshot of a specified publisher's metadata at the time of request. It requires the publisher's ID as part of the URL, which can be found using the GraphiQL API.

The Thoth telescope downloads the ONIX metadata files and then transforms the data into a format suitable for loading into BigQuery with the ONIX parser Java command line tool. This is a near-identical process to how the *ONIX telescope's* data-transformation step is executed. The transformed data is loaded into BigQuery, where it can be picked up and used by the ONIX Workflow.

The corresponding table in BigQuery is `onix.onixYYYYMMDD`.

Summary	
Average runtime	5-10 mins
Average download size	1-10 MB
Harvest Type	URL
Harvest Frequency	Weekly
Runs on remote worker	False
Catchup missed runs	False
Credentials Required	No
Uses Telescope Template	None
Each shard includes all data	Yes

Configuration

The following settings need to be configured for the Thoth telescope.

Telescope API Instance

A Thoth Telescope API instance needs to be created. Unlike the ONIX telescope, it does not require any 'extra' fields.

Airflow Connections

The Thoth telescope does not require any airflow connections to run, as the Thoth API is freely usable.

Latest schema

name	type	mode	description
CountryOfManufacture	STRING	NULLABLE	An ISO code identifying the country of manufacture of a single-item product, or of a multiple-item product when all items are manufactured in the same country. This information is needed in some countries to meet regulatory requirements. Optional and non-repeating.
RecordSourceName	STRING	NULLABLE	The name of the party which issued the record, as free text. Optional and non-repeating, independently of the occurrence of any other field.
RecordSourceType	STRING	NULLABLE	An ONIX description which indicates the type of source which has issued the ONIX record. Optional and non- repeating, independently of the occurrence of any other field.
LCCN	STRING	NULLABLE	Library of Congress Control Number
Collections	RECORD	REPEATED	A bibliographic collection in ONIX 3.0 means a fixed or indefinite number of products, published over a fixed or indefinite time period, which share collective attributes (including a collective title) that are required as part of the bibliographic record of each individual product. In this respect, such a collection is most often thought of as a series. A bibliographic collection may, however, also be traded as a single product (often thought of as a set), but this does not alter the way in which its collective attributes are described in the ONIX records for the individual products.
Collections.TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleElements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.
Collections.TitleDetails.TitleElements.PartNumber	RECORD	NULLABLE	When a title element includes a part designation within a larger whole (eg Part I, or Volume 3), this field should be used to carry the number and its 'caption' as text. Optional and non-repeating.
Collections.TitleDetails.TitleElements.PartNumber.Value	STRING	NULLABLE	PartNumber value.
Collections.TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.
Collections.TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.
Collections.TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
Collections.TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> elements is also present.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Collections.TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Collections.TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.
Collections.TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.
Collections.TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.
Collections.CollectionIdentifers	RECORD	REPEATED	A repeatable group of data elements which together specify an identifier of a bibliographic collection. The composite is optional, and may only repeat if two or more identifiers of different types are sent for the same collection. It is not permissible to have two identifiers of the same type.
Collections.CollectionIdentifers.CollectionIdType	STRING	NULLABLE	An ONIX description identifying a scheme from which an identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.
Collections.CollectionIdentifers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <CollectionIDType> field. Mandatory in each occurrence of the <CollectionIdentifier> composite, and non-repeating.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Collections.CollectionIdentifers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in <CollectionIDType> indicates a proprietary scheme, eg a publisher's own code. Optional and non-repeating.
Collections.CollectionType	STRING	NULLABLE	An ONIX description indicating the type of a collection: publisher collection, ascribed collection, or unspecified. Mandatory in each occurrence of the <Collection> composite, and non-repeating.
EditionNumber	INTEGER	NULLABLE	The number of a numbered edition. Optional and non-repeating. Normally sent only for the second and subsequent editions of a work, but by agreement between parties to an ONIX exchange a first edition may be explicitly numbered.
RecordRef	STRING	NULLABLE	Two mandatory data elements must be included at the beginning of every product record or update. The first, <RecordReference>, is a string of text which uniquely identifies the record. The second, <NotificationType>, is a code which specifies the type of notification or update.
RelatedWorks	RECORD	REPEATED	A group of data elements which together describe a work which has a specified relationship to a content item. Optional and repeatable.
RelatedWorks.WorkRelationCode	STRING	NULLABLE	An ONIX description which identifies the nature of the relationship between a product and a work. Mandatory in each occurrence of the <RelatedWork> composite, and non-repeating.
RelatedWorks.WorkIdentifiers	RECORD	REPEATED	A group of data elements which together define an identifier of a work in accordance with a specified scheme. Mandatory in each occurrence of the <RelatedWork> composite, and repeatable only if two or more identifiers for the same work are sent using different identifier schemes (eg ISTC and DOI).
RelatedWorks.WorkIdentifiers.WorkIDType	STRING	NULLABLE	An ONIX description identifying the scheme from which the identifier in the <IDValue> element is taken. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
RelatedWorks.WorkIdentifiers.IDValue	STRING	NULLABLE	An identifier of the type specified in the <WorkIDType> element. Mandatory in each occurrence of the <WorkIdentifier> composite, and non-repeating.
RelatedWorks.WorkIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <WorkIDType> element indicates a proprietary scheme. Optional and non-repeating.
TextContent	RECORD	REPEATED	An optional group of data elements which together carry text related to the product, repeatable in order to deliver multiple texts (often of different types, though for some text types, there may be multiple instances of that type).
TextContent.TextType	STRING	NULLABLE	An ONIX description which identifies the type of text which is sent in the <Text> element. Mandatory in each occurrence of the <TextContent> composite, and non-repeating.
TextContent.Text	STRING	REPEATED	The text specified in the <TextType> element. Mandatory in each occurrence of the <TextContent> composite, and repeatable when essentially identical text is supplied in multiple languages. The language attribute is optional for a single instance of <Text>, but must be included in each instance if <Text> is repeated.
CityOfPublications	STRING	REPEATED	The name of a city or town associated with the imprint or publisher. Optional, and repeatable if parallel names for a single location appear on the title page in multiple languages, or if the imprint carries two or more cities of publication.
DOI	STRING	NULLABLE	The product's Digital object identifier.
EditionType	STRING	REPEATED	An ONIX description, indicating the type of a version or edition. Optional, and repeatable if the product has characteristics of two or more types (eg 'revised' and 'annotated').
Imprints	RECORD	REPEATED	An optional group of data elements which together identify an imprint or brand under which the product is marketed. The composite must carry either a name identifier or a name or both, and is repeatable to specify multiple imprints or brands.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Imprints.ImprintIdentifiers	RECORD	REPEATED	A group of data elements which together define the identifier of an imprint name. Optional, but mandatory if the <Imprint> composite does not carry an <ImprintName>. The composite is repeatable in order to specify multiple identifiers for the same imprint or brand.
Imprints.ImprintIdentifiers.ImprintIDType	STRING	NULLABLE	An ONIX description which identifies the scheme from which the value in the <IDValue> element is taken. Mandatory in each occurrence of the <ImprintIdentifier> composite.
Imprints.ImprintIdentifiers.IDValue	STRING	NULLABLE	A code value taken from the scheme specified in the <ImprintIDType> element. Mandatory in each occurrence of the <ImprintIdentifier> composite, and non-repeating..
Imprints.ImprintIdentifiers.IDTypeName	STRING	NULLABLE	A name which identifies a proprietary identifier scheme (ie a scheme which is not a standard and for which there is no individual ID type code). Must be used when, and only when, the code in the <ImprintIDType> element indicates a proprietary scheme. Optional and non-repeating.
Imprints.ImprintName	STRING	NULLABLE	The name of an imprint or brand under which the product is issued, as it appears on the product. Mandatory if there is no imprint identifier in an occurrence of the <Imprint> composite, and optional if an imprint identifier is included. Non-repeating.
Publishers	RECORD	REPEATED	An optional group of data elements which together identify an entity which is associated with the publishing of a product. The composite allows additional publishing roles to be introduced without adding new fields. Each occurrence of the composite must carry a publishing role code and either a name identifier or a name or both, and the composite is repeatable in order to identify multiple entities.
Publishers.PublisherName	STRING	NULLABLE	The name of an entity associated with the publishing of a product. Mandatory if there is no publisher identifier in an occurrence of the <Publisher> composite, and optional if a publisher identifier is included. Non-repeating.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Publishers.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the publisher identified in an occurrence of the <Publisher> composite. Repeatable in order to provide links to multiple websites.
Publishers.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Publishers.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.
Publishers.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Publishers.PublishingRole	STRING	NULLABLE	An ONIX description which identifies a role played by an entity in the publishing of a product. Mandatory in each occurrence of the <Publisher> composite, and non-repeating.
RelatedProducts	RECORD	REPEATED	A group of data elements which together describe a product which has a specified relationship to a content item. Optional and repeatable.
RelatedProducts.ISBN13	STRING	NULLABLE	The related product's 13-digit International Standard Book Number.
RelatedProducts.ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
RelatedProducts.DOI	STRING	NULLABLE	The related product's digital object identifier.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
RelatedProducts.GTIN_13	STRING	NULLABLE	The related product's 13-digit global trade item number.
RelatedProducts.ProductRelationCodes	STRING	REPEATED	An ONIX description which identifies the nature of the relationship between two products, eg 'replaced-by'. Mandatory in each occurrence of the <RelatedProduct> composite, and repeatable where the related product has multiple types of relationship to the product described.
RelatedProducts.PID_Proprietary	STRING	NULLABLE	The related product's proprietary product ID.
PID_Proprietary	STRING	NULLABLE	The product's proprietary product identifier.
ISBN10	STRING	NULLABLE	The product's 10-digit International Standard Book Number.
ISBN13	STRING	NULLABLE	The product's 13-digit International Standard Book Number.
TitleDetails	RECORD	REPEATED	A group of data elements which together give the text of a title and specify its type. At least one title detail element is mandatory in each occurrence of the <DescriptiveDetail> composite, to give the primary form of the product title. The composite is repeatable with different title types.
TitleDetails.TitleElements	RECORD	REPEATED	A group of data elements which together represent an element of a title. At least one title element is mandatory in each occurrence of the <TitleDetail> composite. The composite is repeatable with different sequence numbers and/or title element levels, each repeat carrying a different part of the title. An instance of the <TitleElement> composite must include at least one of: <PartNumber>; <YearOfAnnual>; <TitleText>, <NoPrefix/> together with <TitleWithoutPrefix>, or <TitlePrefix> together with <TitleWithoutPrefix>. In other words it must carry either the text of a title or a part or year designation; and it may carry both.
TitleDetails.TitleElements.TitleWithoutPrefix_TextCaseFlags	STRING	NULLABLE	TitleWithoutPrefix textcase attribute.
TitleDetails.TitleElements.TitleText_TextCaseFlags	STRING	NULLABLE	TitleText textcase attribute.
TitleDetails.TitleElements.Subtitle_TextCaseFlags	STRING	NULLABLE	Subtitle textcase attribute.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
TitleDetails.TitleElements.TitleText	STRING	NULLABLE	The text of a title element, excluding any subtitle. Optional and non-repeating, may only be used where <TitlePrefix>, <NoPrefix/> and <TitleWithoutPrefix> are not used.
TitleDetails.TitleElements.TitleText_Language	STRING	NULLABLE	TitleText language attribute.
TitleDetails.TitleElements.TitleElementLevel	STRING	NULLABLE	An ONIX description indicating the level of a title element: collection level, subcollection level, or product level. Mandatory in each occurrence of the <TitleElement> composite, and non-repeating.
TitleDetails.TitleElements.Subtitle	STRING	NULLABLE	The text of a subtitle, if any. 'Subtitle' means any added words which appear with the title element given in an occurrence of the <TitleElement> composite, and which amplify and explain the title element, but which are not considered to be part of the title element itself. Optional and non-repeating.
TitleDetails.TitleElements.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
TitleDetails.TitleElements.TitleWithoutPrefix	STRING	NULLABLE	The text of a title element without the title prefix; and excluding any subtitle. Optional and non-repeating; can only be used if one of the <NoPrefix/> or <TitlePrefix> element is also present.
TitleDetails.TitleElements.Subtitle_Language	STRING	NULLABLE	Language attribute.
TitleDetails.TitleElements.TitlePrefix	RECORD	NULLABLE	Text at the beginning of a title element which is to be ignored for alphabetical sorting. Optional and non- repeating; can only be used when <TitleText> is omitted, and if the <TitleWithoutPrefix> element is also present. These two elements may be used in combination in applications where it is necessary to distinguish an initial word or character string which is to be ignored for filing purposes, eg in library systems and in some bookshop databases.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
TitleDetails.TitleElements.TitlePrefix.Value	STRING	NULLABLE	TitlePrefix value.
TitleDetails.TitleType	STRING	NULLABLE	An ONIX description indicating the type of a title. Mandatory in each occurrence of the <TitleDetail> composite, and non-repeating.
TitleDetails.TitleStatement	STRING	NULLABLE	Free text showing how the overall title (including any collection level title, if the collection title is treated as part of the product title and included in P.6) should be presented in any display, particularly when a standard concatenation of individual title elements from Group P.6 (in the order specified by the <SequenceNumber> data elements) would not give a satisfactory result. Optional and non-repeating. When this field is sent, the recipient should use it to replace all title detail sent in Group P.6 for display purposes only. The individual title element detail must also be sent, for indexing and retrieval purposes.
PublishingDates	RECORD	REPEATED	A group of data elements which together specify a date associated with the publishing of the product. Optional, but where known, at least a date of publication must be specified either here (as a 'global' pub date) or in <MarketPublishingDetail> (P.25). Other dates related to the publishing of a product can be sent in further repeats of the composite
PublishingDates.PublishingDateRole	STRING	NULLABLE	An ONIX description indicating the significance of the date, eg publication date, announcement date, latest reprint date. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating.
PublishingDates.Date	INTEGER	NULLABLE	The date specified in the <PublishingDateRole> field. Mandatory in each occurrence of the <PublishingDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
GTIN_13	STRING	NULLABLE	The product's 13-digit global trade item number.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Languages	RECORD	REPEATED	A group of data elements which together represent a language, and specify its role and, where required, whether it is a country variant. Optional, and repeatable to specify multiple languages and their various roles.
Languages.CountryCode	STRING	NULLABLE	A code identifying the country when this specifies a variant of the language, eg US English. Optional and non-repeating.
Languages.LanguageRole	STRING	NULLABLE	An ONIX description indicating the 'role' of a language in the context of the ONIX record. Mandatory in each occurrence of the <Language> composite, and non-repeating.
Languages.LanguageCode	STRING	NULLABLE	An ISO code indicating a language. Mandatory in each occurrence of the <Language> composite, and non-repeating.
ProductForm	STRING	NULLABLE	An ONIX description which indicates the primary form of a related product. Optional in an occurrence of <RelatedProduct>, and non-repeating. If supplied, should be identical to the <ProductForm> element supplied in the <DescriptiveDetail> block of the full ONIX record describing the related product itself.
Contributors	RECORD	REPEATED	A group of data elements which together describe a personal or corporate contributor to the product. Optional, and repeatable to describe multiple contributors.
Contributors.LettersAfterNames	STRING	NULLABLE	The seventh part of a structured name of a person who contributed to the creation of the product: qualifications and honors following a person's names, eg 'CBE FRS'. Optional and non-repeating.
Contributors.Gender	STRING	NULLABLE	An optional ONIX code specifying the gender of a personal contributor. Not repeatable. Note that this indicates the gender of the contributor's public identity (which may be pseudonymous) based on designations used in ISO 5218, rather than the gender identity, biological sex or sexuality of a natural person.
Contributors.Proprietary	INTEGER	NULLABLE	The contributor's proprietary identifier.
Contributors.NameType	STRING	NULLABLE	An ONIX description indicating the type of a primary name. Optional, and non-repeating. If omitted, the default is 'unspecified'.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Contributors.ProfessionalAffiliations	RECORD	REPEATED	An optional group of data elements which together identify a contributor's professional position and/or affiliation, repeatable to allow multiple positions and affiliations to be specified.
Contributors.ProfessionalAffiliations.Positions	STRING	REPEATED	A professional position held by a contributor to the product at the time of its creation. Optional, and repeatable to provide parallel text in multiple languages. The language attribute is optional for a single instance of <ProfessionalPosition>, but must be included in each instance if <ProfessionalPosition> is repeated.
Contributors.ProfessionalAffiliations.Affiliations	STRING	NULLABLE	An organization to which a contributor to the product was affiliated at the time of its creation, and – if the <ProfessionalPosition> element is also present – where s/he held that position. Optional and non-repeating.
Contributors.ORCID	STRING	NULLABLE	A 16-digit ORCID ID that uniquely identifies the author.
Contributors.BiographicalNotes	RECORD	REPEATED	A biographical note about a contributor to the product. (See the <TextContent> composite in Group P.14 for a biographical note covering all contributors to a product in a single text.) Optional, and repeatable to provide parallel biographical notes in multiple languages. The language attribute is optional for a single instance of <BiographicalNote>, but must be included in each instance if <BiographicalNote> is repeated. May occur with a person name or with a corporate name. A biographical note in ONIX should always contain the name of the person or body concerned, and it should always be presented as a piece of continuous text consisting of full sentences. Some recipients of ONIX data feeds will not accept text which has embedded URLs. A contributor website link can be sent using the <Website> composite below.
Contributors.BiographicalNotes.TextFormat	STRING	NULLABLE	The textformat attribute.
Contributors.BiographicalNotes.Note	STRING	NULLABLE	The biographical note.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Contributors.TitlesBeforeNames	STRING	NULLABLE	The first part of a structured name of a person who contributed to the creation of the product: qualifications and/or titles preceding a person's names, eg 'Professor' or 'HRH Prince' or 'Saint'. Optional and non- repeating: see Group P.7 introductory text for valid options.
Contributors.Roles	STRING	REPEATED	An ONIX description indicating the role played by a person or corporate body in the creation of the product. Mandatory in each occurrence of a <Contributor> composite, and may be repeated if the same person or corporate body has more than one role in relation to the product.
Contributors.Websites	RECORD	REPEATED	An optional group of data elements which together identify and provide a pointer to a website which is related to the person or organization identified in an occurrence of the <Contributor> composite. Repeatable to provide links to multiple websites.
Contributors.Websites.WebsiteDescriptions	STRING	REPEATED	Free text describing the nature of the website which is linked through the <WebsiteLink> element. Optional, and repeatable to provide parallel descriptive text in multiple languages. The language attribute is optional for a single instance of <WebsiteDescription>, but must be included in each instance if <WebsiteDescription> is repeated.
Contributors.Websites.WebsiteRole	STRING	NULLABLE	An ONIX description which identifies the role or purpose of the website which is linked through the <WebsiteLink> element. Optional and non-repeating.
Contributors.Websites.WebsiteLinks	STRING	REPEATED	The URL for the website. Mandatory in each occurrence of the <Website> composite, and repeatable to provide multiple URLs where the website content is available in multiple languages. The language attribute is optional for a single instance of <WebsiteLink>, but must be included in each instance if <WebsiteLink> is repeated.
Contributors.PersonNameInverted	STRING	NULLABLE	The name of a person who contributed to the creation of the product, presented with the element used for alphabetical sorting placed first ('inverted order'). Optional and non-repeating: see Group P.7 introductory text for valid options.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Contributors.Dates	RECORD	REPEATED	A group of data elements which together specify a date associated with the person or organization identified in an occurrence of the <Contributor> composite, eg birth or death. Optional, and repeatable to allow multiple dates to be specified.
Contributors.Dates.Date	INTEGER	NULLABLE	The date specified in the <ContributorDateRole> field. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating. <Date> may carry a dateformat attribute: if the attribute is missing, then <DateFormat> indicates the format of the date; if both dateformat attribute and <DateFormat> element are missing, the default format is YYYYMMDD.
Contributors.Dates.Role	STRING	NULLABLE	An ONIX description indicating the significance of the date in relation to the contributor name. Mandatory in each occurrence of the <ContributorDate> composite, and non-repeating.
Contributors.SequenceNumber	INTEGER	NULLABLE	A number which specifies a single overall sequence of title elements, which is the preferred order for display of the various title elements when constructing a complete title. Optional and non-repeating. It is strongly recommended that where there are multiple title elements within a <TitleDetail> composite, each occurrence of the <TitleElement> composite should carry a <SequenceNumber>.
Contributors.PrefixToKey	STRING	NULLABLE	The third part of a structured name of a person who contributed to the creation of the product: a prefix which precedes the key name(s) but which is not to be treated as part of the key name, eg 'van' in Ludwig van Beethoven. This element may also be used for titles that appear after given names and before key names, eg 'Lord' in Alfred, Lord Tennyson. Optional and non-repeating.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Contributors.KeyNames	STRING	NULLABLE	The fourth part of a structured name of a person who contributed to the creation of the product: key name(s), ie the name elements normally used to open an entry in an alphabetical list, eg 'Smith' or 'Garcia Marquez' or 'Madonna' or 'Francis de Sales' (in Saint Francis de Sales). Non-repeating. Required if name part elements P.7.11 to P.7.18 are used.
Contributors.TitlesAfterNames	STRING	NULLABLE	The eighth part of a structured name of a person who contributed to the creation of the product: titles following a person's names, eg 'Duke of Edinburgh'. Optional and non-repeating.
Contributors.AlternativeNames	STRING	REPEATED	A group of data elements which together represent an alternative name of a contributor, and specify its type. The <AlternativeName> composite is optional, and is repeatable to provide multiple alternative names for the contributor.
Contributors.NamesBeforeKey	STRING	NULLABLE	The second part of a structured name of a person who contributed to the creation of the product: name(s) and/or initial(s) preceding a person's key name(s), eg James J. Optional and non-repeating.
Contributors.Places	RECORD	REPEATED	An optional group of data elements which together identify a geographical location with which a contributor is associated, used to support 'local interest' promotions. Repeatable to identify multiple geographical locations, each usually with a different relationship to the contributor.
Contributors.Places.CountryCode	STRING	NULLABLE	A code identifying a country with which a contributor is particularly associated. Optional and non-repeatable. There must be an occurrence of either the <CountryCode> or the <RegionCode> elements in each occurrence of <ContributorPlace>.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Contributors.Places.Locations	STRING	REPEATED	The name of a city or town location within the specified country or region with which a contributor is particularly associated. Optional, and repeatable to provide parallel names for a single location in multiple languages (eg Baile Átha Cliath and Dublin, or Bruxelles and Brussel). The language attribute is optional for a single instance of <LocationName>, but must be included in each instance if <LocationName> is repeated.
Contributors.Places.Relation	STRING	NULLABLE	An ONIX description identifying the relationship between a contributor and a geographical location. Mandatory in each occurrence of <ContributorPlace> and non-repeating.
Contributors.PersonName	STRING	NULLABLE	The name of a person who contributed to the creation of the product, unstructured, and presented in normal order. Optional and non-repeating: see Group P.7 introductory text for valid options.
Contributors.ISNI	STRING	NULLABLE	16-digit International Standard Name Identifier number.
Contributors.CorprorateName	STRING	NULLABLE	The name of a corporate body which contributed to the creation of the product, unstructured. Optional and non-repeating: see Group P.7 introductory text for valid options.
COKI_ID	STRING	NULLABLE	The product's internal COKI identifier.
Subjects	RECORD	REPEATED	An optional and repeatable group of data elements which together specify a subject classification or subject heading.
Subjects.SubjectHeadingText	STRING	REPEATED	The text of a subject heading taken from the scheme specified in the <SubjectSchemeIdentifier> element, or of free language keywords if the scheme is specified as 'keywords'; or the text equivalent to the <SubjectCode> value, if both code and text are sent. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite.
Subjects.SubjectSchemeIdentifier	STRING	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.
Subjects.SubjectSchemeVersion	FLOAT	NULLABLE	A number which identifies a version or edition of the subject scheme specified in the associated <SubjectSchemeIdentifier> element. Optional and non-repeating.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Subjects.SubjectSchemeName	STRING	NULLABLE	A name identifying a proprietary subject scheme (ie a scheme which is not a standard and for which there is no individual identifier code) when <SubjectSchemeIdentifier> is coded '24'. Optional and non-repeating.
Subjects.SubjectCode	STRING	NULLABLE	A subject class or category code from the scheme specified in the <SubjectSchemeIdentifier> element. Either <SubjectCode> or <SubjectHeadingText> or both must be present in each occurrence of the <Subject> composite. Non-repeating.
Subjects.MainSubject	BOOLEAN	NULLABLE	An empty element that identifies an instance of the <Subject> composite as representing the main subject category for the product. The main category may be expressed in more than one subject scheme, ie there may be two or more instances of the <Subject> composite, using different schemes, each carrying the <MainSubject/> flag, so long as there is only one main category per scheme. Optional and non-repeating in each occurrence of the <Subject> composite.
Extent	RECORD	REPEATED	A group of data elements which together describe an extent pertaining to the product. Optional, but in practice required for most products, eg to give the number of pages in a printed book or paginated e-book, or to give the running time of an audiobook. Repeatable to specify different extent types or units.
Extent.ExtentType	STRING	NULLABLE	An ONIX description which identifies the type of extent carried in the composite, eg running time for an audio or video product. Mandatory in each occurrence of the <Extent> composite, and non-repeating. From Issue 9 of the code lists, an extended set of values for <ExtentType> has been defined to allow more accurate description of pagination.

continues on next page

Table 12 – continued from previous page

name	type	mode	description
Extent.ExtentValue	INTEGER	NULLABLE	The numeric value of the extent specified in <ExtentType>. Optional, and non-repeating. However, either <ExtentValue> or <ExtentValueRoman> must be present in each occurrence of the <Extent> composite; and it is very strongly recommended that <ExtentValue> should always be included, even when the original product uses Roman numerals.
Extent.ExtentUnit	STRING	NULLABLE	An ONIX description indicating the unit used for the <ExtentValue> and the format in which the value is presented. Mandatory in each occurrence of the <Extent> composite, and non-repeating.
Extent.ExtentValueRoman	STRING	NULLABLE	The value of the extent expressed in Roman numerals. Optional, and non-repeating. Used only for page runs which are numbered in Roman.

UCL Discovery

UCL Discovery is UCL's open access repository, showcasing and providing access to the full texts of UCL research publications.

The Google Sheet

UCL's titles are referenced via their identifier - the eprint ID. Their metadata maps the eprint ID to an ISBN13, but not consistently. For this reason, we forgo the use of their metadata and instead employ a semi-manual process to reliably map the two identifiers. The telescope references a Google sheet that contains all of the titles available in the UCL Discovery repository under the following headings:

| Heading | Description | | ———————- | ——————————— | | ISBN13 | The title's ISBN13 | | date | The date of publication | | title_list_title | The title of the publication | | discovery_eprint_id | The eprint ID of the publication |

Some notes :

- These headings are hardcoded into the telescope. Any change in the sheet will break the telescope without prior intervention.
- Entries without a publication date or with a publication date in the future (where the current time is determined by the airflow scheduler) will be ignored.
- Entries missing either an ISBN13 or eprint ID will be ignored.

For the aforementioned reasons, it is important that **the google sheet remain up to date**. Otherwise, the usage for a title may be missed and require a rerun.

Access

Access to the sheet can be granted using the sheet UI (*Share* at the top right of the page). The telescope will access the sheet via a service account, which will need to be given read access (*Viewer*) by supplying the account's email address.

Usage API

UCL Discovery provides free and open access to their usage REST API. Unfortunately, I can't find any documentation on its use and design. We utilise two endpoints:

- **Countries URI** = https://discovery.ucl.ac.uk/cgi/stats/get?from=[YYYYMMDD]&to=[YYYYMMDD]&irs2report=eprint&set_n
- **Totals URI** = https://discovery.ucl.ac.uk/cgi/stats/get?from=[YYYYMMDD]&to=[YYYYMMDD]&irs2report=eprint&set_name

Where *from*, *to* and *set_value* are appropriately set. The countries URI returns statistics pertaining to the number of downloads of the provided eprint ID broken down by country. The totals URI returns statistics pertaining to the number of downloads of the provided eprint ID aggregated over all regions. It should be noted that the *totals* data is not necessarily a simply aggregation of the *countries* data. This is because country data is omitted for downloads that are not attributed to a region. It is therefore not uncommon to have a total download count (derived from the totals URI) that is greater than the sum of all downloads from all listed countries (from the countries URI).

Telescope Workflow

The telescope's workflow can be broken down as such:

Download

Acquires the eprint IDs and publication dates from *the Google Sheet*. For each ID that has a publication date that is before the current scheduled run date, download the country and totals data. Then upload to GCS download bucket.

Transform

Acquires the eprint IDs, ISBN13s and titles from *the Google Sheet*. For each ID, load the downloaded data (both coutried and totals) into a single data structure and include the title (whether it is empty or not does not matter - the title exists for completeness only). Add an additional field to each row - the *release_date* which is determined by the scheduled runtime. Upload this transformed structure to GCS transform bucket.

BQ Load

Load the table into BigQuery and partition on the *release_date*.

Run Summary

The corresponding table in BigQuery is `ucl.ucl_discoveryYYYYMMDD`.

Summary	
Average runtime	2 min
Average download size	1.5 MB
Harvest Type	API
Harvest Frequency	Monthly
Runs on remote worker	False
Catchup missed runs	True
Table Write Disposition	Append
Update Frequency	Daily
Credentials Required	No
Each shard includes all data	No

Latest schema

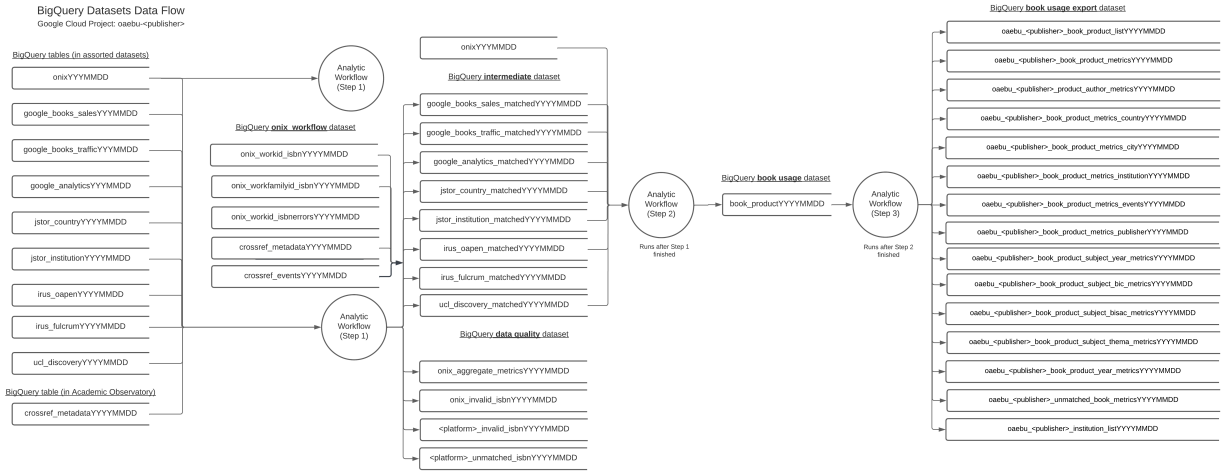
name	type	mode	description
ISBN	STRING	REQUIRED	ISBN13 of the book.
eprint_id	STRING	REQUIRED	eprint ID of the book.
title	STRING	NULLABLE	Title of the book.
timescale	RECORD	NULLABLE	Timescale of the statistics as reported by the origin.
timescale.format	STRING	NULLABLE	Format of the 'to' and 'from' fields
timescale.from	STRING	NULLABLE	Beginning of date range for the statistics
timescale.to	STRING	NULLABLE	End of date range for the statistics
origin	RECORD	NULLABLE	Origin of the statistics
origin.url	STRING	NULLABLE	The URL of the origin
origin.name	STRING	NULLABLE	The name of the origin
total_downloads	INTEGER	NULLABLE	The aggregated statistics for the reported period
country	RECORD	REPEATED	The aggregated statistics for each reported country
country.value	STRING	NULLABLE	The two letter country code.
country.count	INTEGER	NULLABLE	The total number of item downloads for the reported period from this country.
release_date	DATE	REQUIRED	Last day of the release month. Table is partitioned on this column.

3.1.2 Analytic workflows

Analytic workflows process the data ingested by telescope workflows and are also built on top of Apache Airflow DAGs.

Analytic workflows

Analytic workflows process the data ingested by telescope workflows and are also built on top of Apache Airflow DAGs.

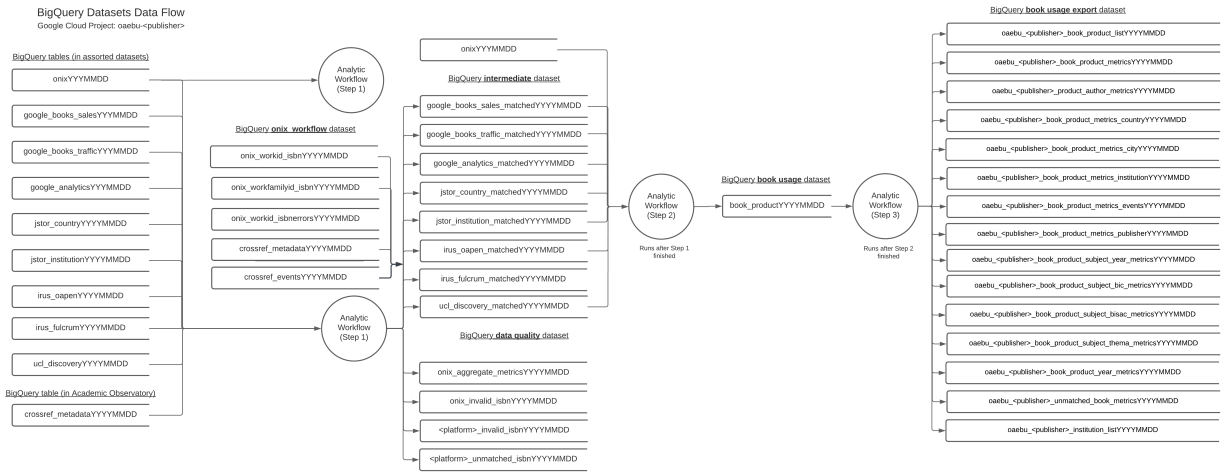


The pilot project has one core workflow, which is broken into three parts and described below. The parts are:

1. Aggregating and Mapping book products into works and work families
2. Linking data from metric providers to book products
3. Creating export tables for visualisation in dashboards

Analytic workflows introduction

Analytic workflows process the data ingested by telescope workflows and are also built on top of Apache Airflow DAGs.



The pilot project has one core workflow, which is broken into three parts and described below. The parts are:

1. Aggregating and Mapping book products into works and work families
2. Linking data from metric providers to book products
3. Creating export tables for visualisation in dashboards

ONIX workflow Step 1 - Mapping Book Products

The ONIX workflow uses the ONIX table created by the ONIX telescope to do the following:

1. Aggregate book product records into works records. Works are equivalence classes of products, where each product in the class is a manifestation of each other. For example, a PDF and a paperback of the same work.
2. Aggregate work records into work family records. A work family is an equivalence class of works where each work in the class is just a different edition.
3. Produce intermediate lookup tables mapping ISBN13 -> WorkID and ISBN13 -> WorkFamilyID.
4. Produce intermediate tables that append work_id and work_family_id columns to different data tables with ISBN keys.

Link to Query

Definitions - Product, Work and Work Families

A **Product**: A product is a manifestation of a work, and will have its own ISBN. There may be several DOIs linked to a single product though (or sometimes none at all).

A **Work**: Can be a collection of products, which are each different manifestation of the same work. Some datasets have unique IDs assigned to the concept of a work, but these are not as clear as the usage of ISBN for a product.

An **Edition**: Is a new Work, but is derived as a revision from an existing work as opposed to being entirely new.

A **Work Family** is a collection of works which are different editions of each other.

Dependencies

The ONIX workflow is dependent on the ONIX telescope. It waits for the ONIX telescope to finish before it starts executing. This requires an ONIX telescope to be present and scheduled.

Link to Code

Work ID

The Work ID will be an arbitrary ISBN representative from a product in the equivalence class.

name	type	mode	description
isbn13	STRING	NULLABLE	ISBN13
work_id	STRING	NULLABLE	The WorkID. Likely to be an ISBN.

Work Family ID

The Work Family ID will be an arbitrary Work ID (ISBN) representative from a work in the equivalence class.

name	type	mode	description
isbn13	STRING	NULLABLE	ISBN13
work_family_id	STRING	NULLABLE	The Work Family ID. Likely to be an ISBN.

ONIX Work ID ISBN Errors

name	type	mode	description
Error	STRING	NULLABLE	Error string

Create Crossref metadata table

Crossref Metadata is required to proceed. The ISBNs for each work is obtained from the publisher's Onix table. For each of these ISBNs, the Crossref Metadata table produced by the Academic Observatory workflows is queried. Refer to the Crossref Metadata telescope.

Create Crossref events table

Similarly to Crossref Metadata, Crossref Event Data is retrieved through Crossref's dedicated event REST API through the Crossref Event Data telescope. The API accepts queries based on DOI only, which we retrieve by matching the appropriate ISBN13 from the metadata.

Create book table

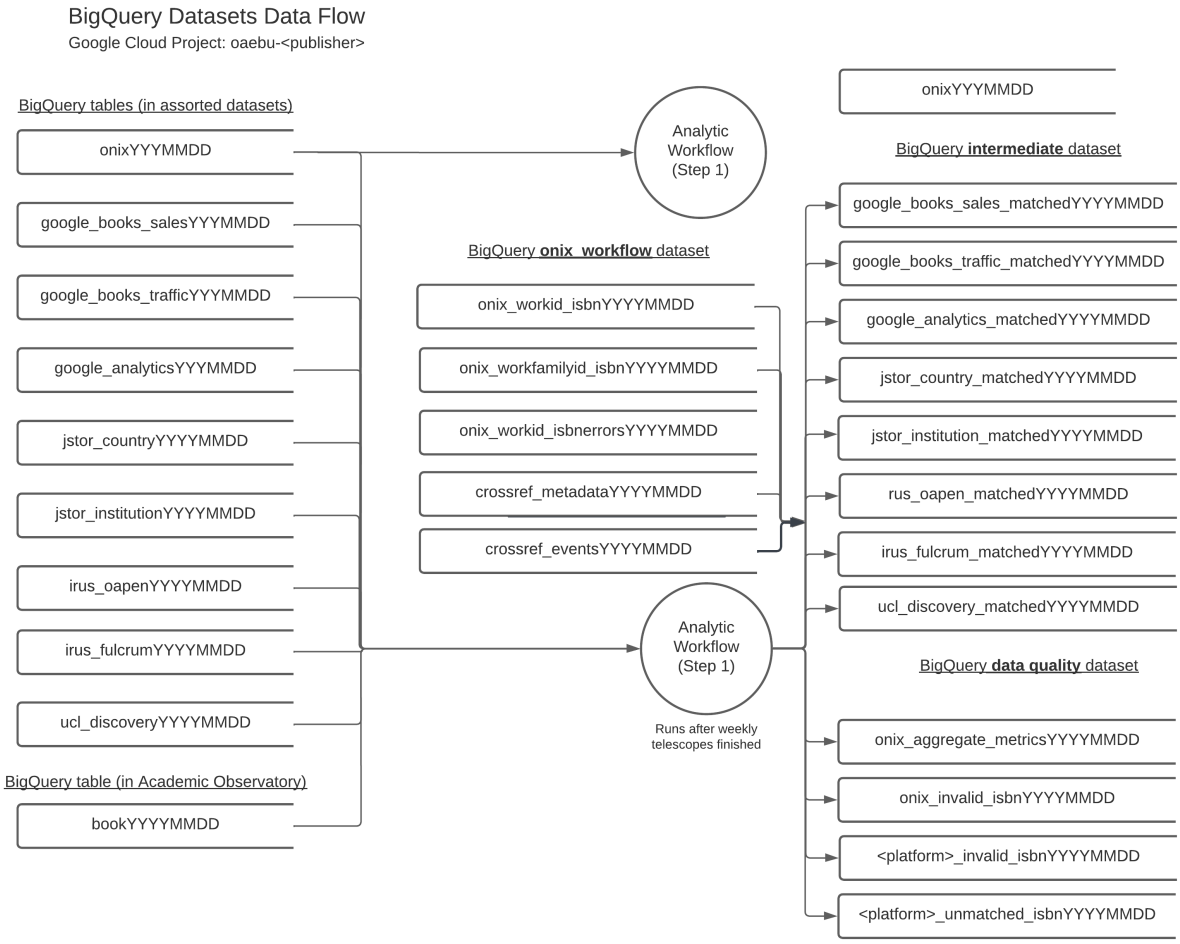
The book table is a collection of works and their relevant details for the relative publisher. The table accommodates a title's Crossref metadata, events and separate chapters.

Create intermediate tables

For each data partner's tables containing ISBN, create new "matched" tables which extend the original data with new "work_id" and "work_family_id" columns.

The schemas for these tables are identical to the raw Telescope's schemas, with the addition of work_ids and work_family_ids.

Link to Query



Create QA tables

For each data source, including the intermediate tables, we perform basic quality assurance checks on the data, and output the results to tables that are easy to export for analysis by the publisher (e.g. to CSV). For example we verify if the provided ISBNs are valid, or if there are unmatched ISBNs indicating that there are missing ONIX product records.

ONIX Aggregate Metrics

Link to Query

name	type	mode	description
table_size	INTEGER	NULLABLE	Total Number of Book Products
no_isbns	INTEGER	NULLABLE	Count of how many rows are missing an ISBN
no_relatedworks	INTEGER	NULLABLE	Count of how many rows are a related work
no_relatedproducts	INTEGER	NULLABLE	Count of how many rows are a related product

continues on next page

Table 17 – continued from previous page

name	type	mode	description
no_doi	INTEGER	NULLABLE	Count of how many rows are missing a DOI
no_productform	INTEGER	NULLABLE	Count of how many rows are missing a product form
no_contributors	INTEGER	NULLABLE	Count of how many rows are missing contributors
no_titledetails	INTEGER	NULLABLE	Count of how many rows are missing title details
no_publisher_urls	INTEGER	NULLABLE	Count of how many rows are missing a publisher url

ONIX Invalid ISBN

Details ISBN13s in the ONIX feed that are not valid.

name	type	mode	description
ISBN13	STRING	NULLABLE	ISBN13

Data Platform Invalid ISBN

Details ISBN13s in the data source that are not valid. An example schema is below, as data platforms may use different name fields (e.g, 'ISBN', 'publication_id', 'Primary_ISBN').

name	type	mode	description
ISBN	STRING	NULLABLE	ISBN13

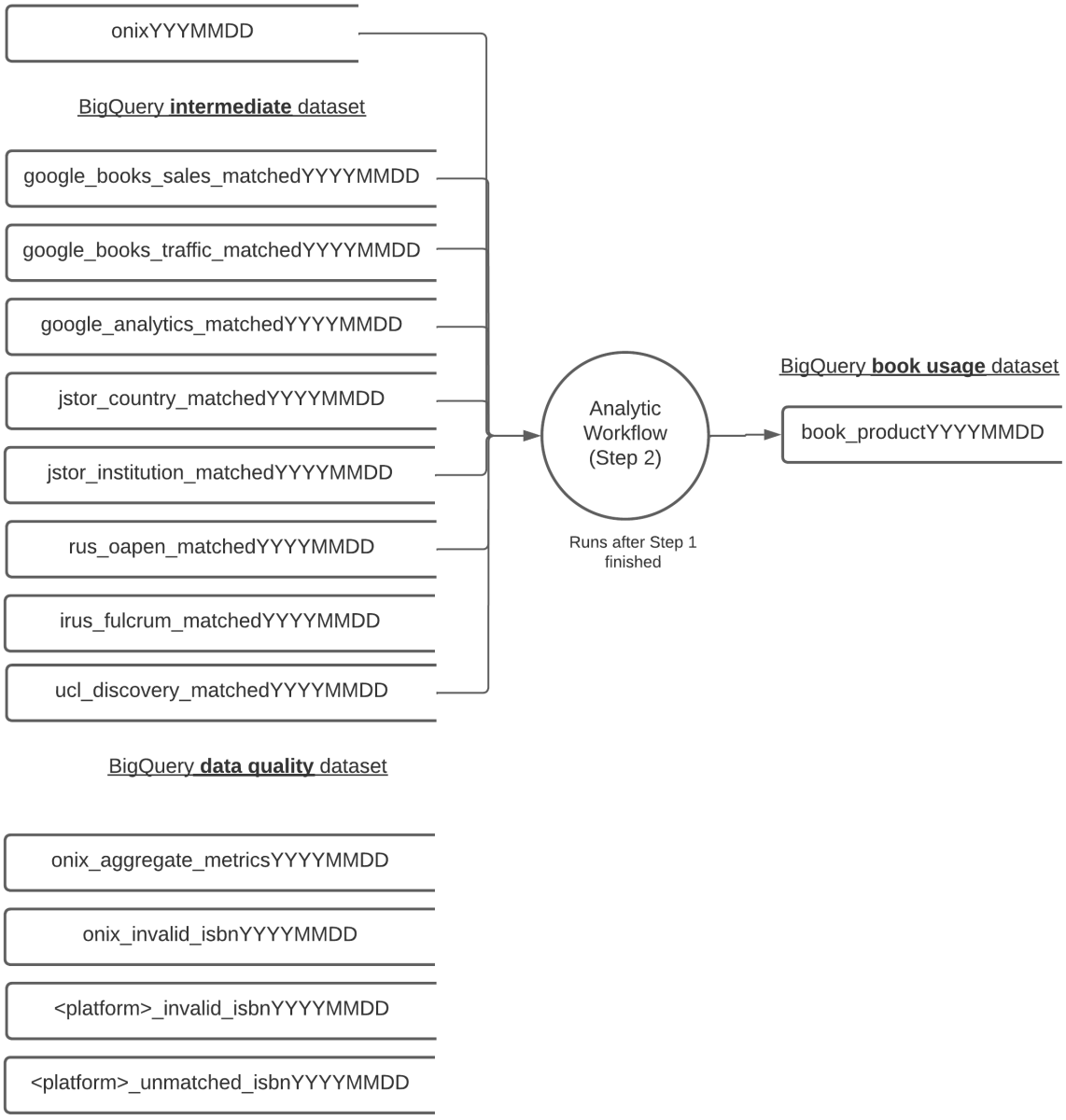
Data Platform Unmatched ISBN

Details ISBN-13s in the data source that were not matched to ISBN-13s in the ONIX feed.

name	type	mode	description
ISBN	STRING	NULLABLE	ISBN13
title	STRING	NULLABLE	Book title from the data source
release_date	DATE	NULLABLE	The release date (month)

ONIX workflow Step 2 - Linking Metrics

Step 2 of the ONIX workflow takes the metrics fetched through various telescopes, then aggregates and joins them to the book records in the publisher's ONIX feed.



Book Product Schema

The output is the book_product table, containing one row per unique book, with a nested month field, which groups all the metrics relating to that book for each calendar month.

Link to Query

name	type	mode	description
ISBN13	STRING	NULLABLE	ISBN13
onix	RECORD	NULLABLE	Fields Pulled from the ONIX Record for this Book Product
onix.Doi	STRING	NULLABLE	DOI
onix.ProductForm	STRING	NULLABLE	The product form, such as digital, print etc
onix.EditionNumber	INTEGER	NULLABLE	The edition number of this book product
onix.title	STRING	NULLABLE	The Book's Title
onix.published_year	STRING	NULLABLE	The year the book was published
onix.published_date	DATE	NULLABLE	The date the book was published
onix.bic_subjects	STRING	REPEATED	A list of BIC subjects
onix.bisac_subjects	STRING	REPEATED	A list of BISAC subjects
onix.thema_subjects	STRING	REPEATED	A list of THEMA subjects
onix.keywords	STRING	REPEATED	A list of Keywords
onix.authors	RECORD	REPEATED	Book Authors
onix.authors.PersonName	STRING	NULLABLE	The Author's Full Name in the format '[first name] [last name]'
onix.authors.PersonNameInverted	STRING	NULLABLE	The Authors Full Name in the format '[last name], [first name]'
onix.authors.ORCID	STRING	NULLABLE	Authors ORCID ID, if present
work_id	STRING	NULLABLE	The dervied Work_ID that we calculate
work_family_id	STRING	NULLABLE	The Dervied Work_Family_ID that we calculate
metadata	RECORD	NULLABLE	Metadata on this book, derived and organised by source
metadata.crossref_objects	RECORD	REPEATED	Linked Objects from Crossref and their values
metadata.crossref_objects.doi	STRING	NULLABLE	The DOI from crossref
metadata.crossref_objects.title	STRING	REPEATED	The title from crossref
metadata.crossref_objects.type	STRING	NULLABLE	The type from crossref
metadata.crossref_objects.publisher	STRING	NULLABLE	The publisher from crossref
metadata.crossref_objects.published_year	INTEGER	NULLABLE	The published year from crossref
metadata.crossref_objects.published_year_month	STRING	NULLABLE	The published year-month from crossref
metadata.crossref_objects.work_isbns	STRING	REPEATED	ISBNs
metadata.chapters	RECORD	REPEATED	Linked Objects from Crossref where they are of type book-chapter only
metadata.chapters.doi	STRING	NULLABLE	The Book Chapter DOI
metadata.chapters.title	STRING	REPEATED	The Book Chapter title
metadata.chapters.type	STRING	NULLABLE	The Book Chapter type
metadata.events	RECORD	REPEATED	Count of events from Crossref Events
metadata.events.source	STRING	NULLABLE	Event Source Type

continues on next page

Table 21 – continued from previous page

name	type	mode	description
metadata.events.count	INTEGER	NULLABLE	Count of events
metadata.google_books_sales	RECORD	NULLABLE	Metadata derived from Google Books Sales
metadata.google_books_sales.ISBN13	STRING	NULLABLE	ISBN
metadata.google_books_sales.Imprint_Name	STRING	NULLABLE	The template used for the book.
metadata.google_books_sales.Title	STRING	NULLABLE	The title of the book.
metadata.google_books_sales.Author	STRING	NULLABLE	The author of the book.
metadata.google_books_traffic	RECORD	NULLABLE	Metadata derived from Google Books Sales
metadata.google_books_traffic.ISBN13	STRING	NULLABLE	ISBN
metadata.google_books_traffic.Title	STRING	NULLABLE	The title of the book
metadata.jstor_metadata	RECORD	NULLABLE	Metadata derived from JSTOR
metadata.jstor_metadata.ISBN13	STRING	NULLABLE	ISBN of the book (13 digits)
metadata.jstor_metadata.Book_Title	STRING	NULLABLE	Title of the book
metadata.jstor_metadata.Book_ID	STRING	NULLABLE	DOI of the book on JSTOR
metadata.jstor_metadata.Authors	STRING	NULLABLE	Author of the book
metadata.jstor_metadata.ISBN	STRING	NULLABLE	ISBN of the book
metadata.jstor_metadata.eISBN	STRING	NULLABLE	ISBN of the digital version of the book (13 digits)
metadata.jstor_metadata.Copyright_Year	INTEGER	NULLABLE	Publication year
metadata.jstor_metadata.Disciplines	STRING	NULLABLE	Subject category of the book
metadata.jstor_metadata.Usage Type	STRING	NULLABLE	For our case it is Open Access
metadata.jstor_institution_metadata	RECORD	NULLABLE	Metadata derived from JSTOR Institutions
metadata.jstor_institution_metadata.ISBN13	STRING	NULLABLE	ISBN of the book (13 digits)
metadata.jstor_institution_metadata.Book_Title	STRING	NULLABLE	Title of the book
metadata.jstor_institution_metadata.Book_ID	STRING	NULLABLE	DOI of the book on JSTOR
metadata.jstor_institution_metadata.Authors	STRING	NULLABLE	
metadata.jstor_institution_metadata.ISBN	STRING	NULLABLE	ISBN of the book (13 digits)
metadata.jstor_institution_metadata.eISBN	STRING	NULLABLE	ISBN of the digital version of the book (13 digits)
metadata.jstor_institution_metadata.Copyright_Year	INTEGER	NULLABLE	Publication year

continues on next page

Table 21 – continued from previous page

name	type	mode	description
metadata.jstor_institution_metadata.Disciplines	STRING	NULLABLE	Subject category of the book
metadata.jstor_institution_metadata.Usage_Type	STRING	NULLABLE	For our case it is Open Access
metadata.irus_oapen_metadata	RECORD	NULLABLE	Metadata derived from IRUS OAPEN
metadata.irus_oapen_metadata.ISBN13	STRING	NULLABLE	ISBN of the book
metadata.irus_oapen_metadata.book_title	STRING	NULLABLE	Title of the book
metadata.irus_oapen_metadata.publisher	STRING	NULLABLE	The publisher
metadata.irus_fulcrum_metadata	RECORD	NULLABLE	Metadata derived from IRUS Fulcrum
metadata.irus_fulcrum_metadata.ISBN13	STRING	NULLABLE	ISBN of the book
metadata.irus_fulcrum_metadata.book_title	STRING	NULLABLE	Title of the book
metadata.irus_fulcrum_metadata.publisher	STRING	NULLABLE	The publisher
metadata.ucl_discovery_metadata	RECORD	NULLABLE	Metadata derived from UCL Discovery
metadata.ucl_discovery_metadata.ISBN13	STRING	NULLABLE	ISBN of the book
metadata.ucl_discovery_metadata.eprint_id	STRING	NULLABLE	The UCL Discovery eprint ID
metadata.worldreader_metadata	RECORD	NULLABLE	Metadata derived from Worldreader
metadata.worldreader_metadata.ISBN13	STRING	NULLABLE	ISBN of the book
metadata.worldreader_metadata.book_title	STRING	NULLABLE	The title of the book
metadata.internet_archive_metadata	RECORD	NULLABLE	Metadata derived from Internet Archive
metadata.internet_archive_metadata.ISBN13	STRING	NULLABLE	ISBN of the book
metadata.internet_archive_metadata.book_title	STRING	NULLABLE	The title of the book
months	RECORD	REPEATED	Linked Metrics from all sources, organised by month of occurance
months.month	DATE	NULLABLE	Month of Recorded Metrics
months.crossref_events	RECORD	REPEATED	Metrics Derived From Crossref Events
months.crossref_events.source	STRING	NULLABLE	The event source
months.crossref_events.count	INTEGER	NULLABLE	The count of events
months.google_analytics	RECORD	NULLABLE	Metrics derived from Google Analytics
months.google_analytics.views_total_country	RECORD	REPEATED	The total number of views per country
months.google_analytics.views_total_country.name	STRING	NULLABLE	The country name
months.google_analytics.views_total_country.value	INTEGER	NULLABLE	The total number of views
months.google_analytics.downloads_total_country	RECORD	REPEATED	The total number of downloads per country

continues on next page

Table 21 – continued from previous page

name	type	mode	description
months.google_analytics.downloads_total_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_total_country.value	INTEGER	NULLABLE	The total number of downloads
months.google_analytics.downloads_pdf_book_country	RECORD	REPEATED	PDF book downloads per country
months.google_analytics.downloads_pdf_book_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_pdf_book_country.value	INTEGER	NULLABLE	The total number of PDF book downloads
months.google_analytics.downloads_pdf_chapter_country	RECORD	REPEATED	PDF chapter downloads per country
months.google_analytics.downloads_pdf_chapter_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_pdf_chapter_country.value	INTEGER	NULLABLE	The total number of PDF chapter downloads
months.google_analytics.downloads_html_book_country	RECORD	REPEATED	HTML book downloads per country
months.google_analytics.downloads_html_book_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_html_book_country.value	INTEGER	NULLABLE	The total number of HTML book downloads
months.google_analytics.downloads_html_chapter_country	RECORD	REPEATED	HTML chapter downloads per country
months.google_analytics.downloads_html_chapter_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_html_chapter_country.value	INTEGER	NULLABLE	The total number of HTML chapter downloads
months.google_analytics.downloads_epub_book_country	RECORD	REPEATED	EPUB book downloads per country
months.google_analytics.downloads_epub_book_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_epub_book_country.value	INTEGER	NULLABLE	The total number of EPUB book downloads
months.google_analytics.downloads_epub_chapter_country	RECORD	REPEATED	EPUB chapter downloads per country
months.google_analytics.downloads_epub_chapter_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_epub_chapter_country.value	INTEGER	NULLABLE	The total number of EPUB chapter downloads
months.google_analytics.downloads_mobi_book_country	RECORD	REPEATED	MOBI book downloads per country
months.google_analytics.downloads_mobi_book_country.name	STRING	NULLABLE	The country name
months.google_analytics.downloads_mobi_book_country.value	INTEGER	NULLABLE	The total number of MOBI book downloads
months.google_analytics.downloads_mobi_chapter_country	RECORD	REPEATED	MOBI chapter downloads per country
months.google_analytics.downloads_mobi_chapter_country.name	STRING	NULLABLE	The country name

continues on next page

Table 21 – continued from previous page

name	type	mode	description
months.google_analytics.downloads_mobi_chapter_country.value	INTEGER	NULLABLE	The total number of MOBI chapter downloads
months.google_books_sales	RECORD	NULLABLE	Metrics derived from Google Books Sales
months.google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
months.google_books_sales.countries	RECORD	REPEATED	The list of countries where buyers brought the book
months.google_books_sales.countries.Country_of_Sale	STRING	NULLABLE	The country where the buyer bought the book
months.google_books_sales.countries.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
months.google_books_traffic	RECORD	NULLABLE	Metrics derived from Google Books Traffic
months.google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
months.google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
months.google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
months.google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
months.google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
months.google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period). If a user views the same page of your book twice during a session, only a single page view is registered
months.jstor_country	RECORD	REPEATED	Metrics derived from JSTOR Country
months.jstor_country.Country_name	STRING	NULLABLE	Country Name
months.jstor_country.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made from that specific country
months.jstor_institution	RECORD	REPEATED	Metrics derived from JSTOR Institutions
months.jstor_institution.Institution	STRING	NULLABLE	Institution name
months.jstor_institution.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made from that specific institution
months.irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform

continues on next page

Table 21 – continued from previous page

name	type	mode	description
months.irus_oapen.version	STRING	NULLABLE	Version of the OAPEN IRUS UK API, corresponds to the COUNTER report version
months.irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
months.irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
months.irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
months.irus_oapen.unique_iten investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
months.irus_oapen.unique_iten requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
months.irus_oapen.country	RECORD	REPEATED	Record to store statistics on the country level
months.irus_oapen.country.name	STRING	NULLABLE	The country name of the client registered by oapen irus uk
months.irus_oapen.country.code	STRING	NULLABLE	The country code of the client registered by oapen irus uk
months.irus_oapen.country.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
months.irus_oapen.country.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
months.irus_oapen.country.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
months.irus_oapen.country.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
months.irus_oapen.country.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
months.irus_oapen.locations	RECORD	REPEATED	Record to store statistics on the location level
months.irus_oapen.locations.latitude	FLOAT	NULLABLE	The latitude geolocated from the client's ip address
months.irus_oapen.locations.longitude	FLOAT	NULLABLE	The longitude geolocated from the client's ip address
months.irus_oapen.locations.city	STRING	NULLABLE	The city geolocated from the client's ip address
months.irus_oapen.locations.country_name	STRING	NULLABLE	The country name geolocated from the client's ip address
months.irus_oapen.locations.country_code	STRING	NULLABLE	The country code geolocated from the client's ip address
months.irus_oapen.locations.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
months.irus_oapen.locations.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
months.irus_oapen.locations.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
months.irus_oapen.locations.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01

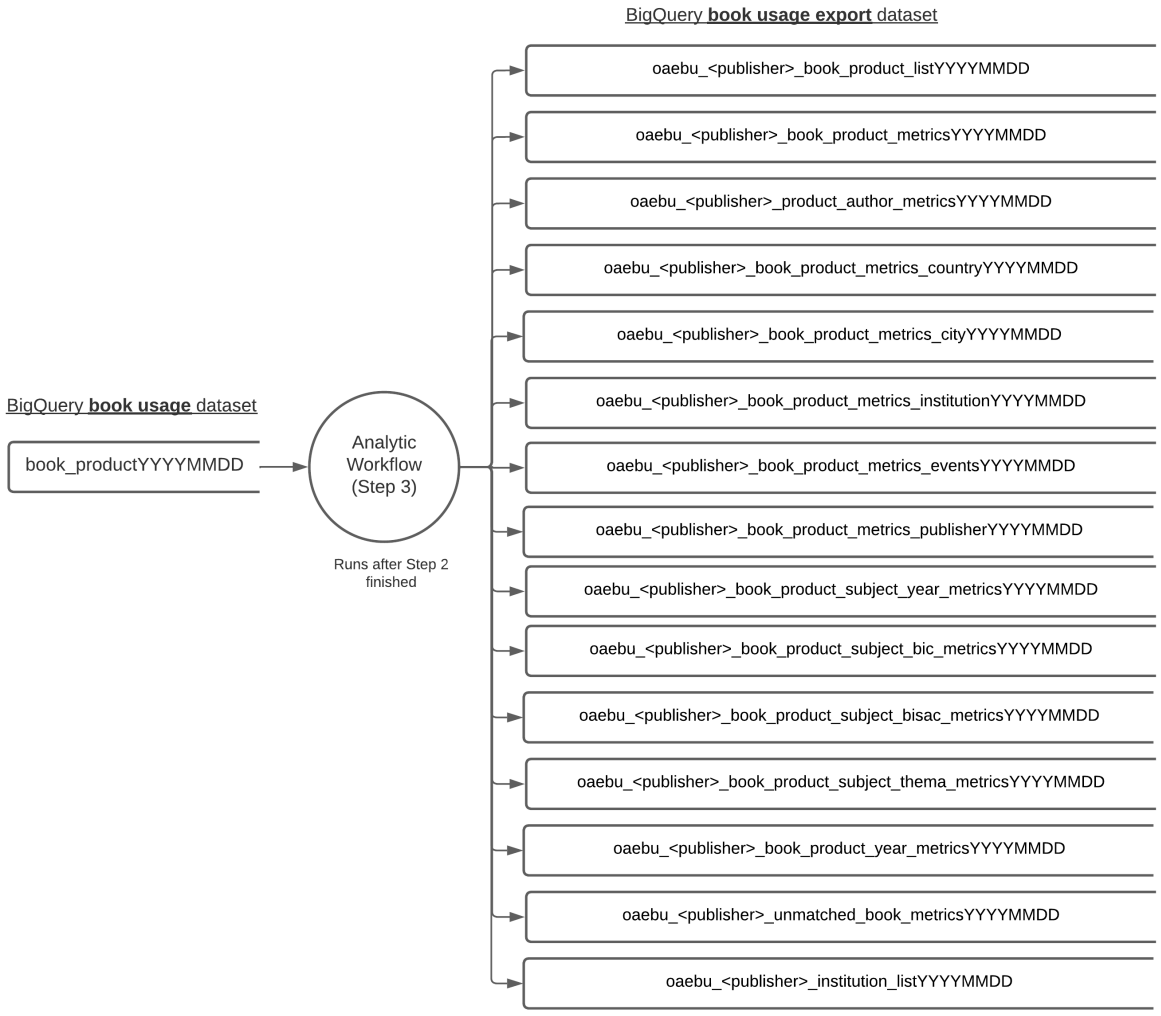
continues on next page

Table 21 – continued from previous page

name	type	mode	description
months.irus_oapen.locations.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
months.irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
months.irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations
months.irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests
months.irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations
months.irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests
months.irus_fulcrum.country	RECORD	REPEATED	Record to store statistics on the country level
months.irus_fulcrum.country.name	STRING	NULLABLE	The country name of the client
months.irus_fulcrum.country.code	STRING	NULLABLE	The country code of the client
months.irus_fulcrum.country.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations
months.irus_fulcrum.country.total_item_requests	INTEGER	NULLABLE	The total number of item requests
months.irus_fulcrum.country.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations
months.irus_fulcrum.country.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests
months.ucl_discovery	RECORD	NULLABLE	Metrics derived from UCL Discovery
months.ucl_discovery.total_downloads	INTEGER	NULLABLE	Number of downloads
months.ucl_discovery.country	RECORD	REPEATED	Number of downloads per country
months.ucl_discovery.country.country_code	STRING	NULLABLE	Country code
months.ucl_discovery.country.country_name	STRING	NULLABLE	Country name
months.ucl_discovery.country.country_downloads	INTEGER	NULLABLE	Number of downloads for the given country
months.worldreader	RECORD	NULLABLE	Metrics derived from Worldreader
months.worldreader.total_downloads	INTEGER	NULLABLE	Number of downloads
months.worldreader.country	RECORD	REPEATED	Number of downloads per country
months.worldreader.country.country_code	STRING	NULLABLE	Country code
months.worldreader.country.country_name	STRING	NULLABLE	Country name
months.worldreader.country.downloads	INTEGER	NULLABLE	Number of downloads for the given country
months.internet_archive	RECORD	NULLABLE	Metrics derived from Internet Archive
months.internet_archive.total_downloads	INTEGER	NULLABLE	Number of downloads

ONIX workflow Step 3 - Exporting to Looker Studio

Step three of the ONIX workflow is to export the book_product table to a sequence of flattened data export tables. The data in these tables is not materially different to the book product table, just organised in a way better suited for dashboards in Looker Studio.



Since these are date-sharded tables, their names will be updated each time the workflow is run. When using Google's Looker (previously Data Studio), it is preferable for us to use a static naming scheme. For this reason, after creating the (sharded) *export* and *quality analysis* tables, we also create/update a *view* for table. These views have a static name. By referencing the view, we can keep the Looker dashboards up-to-date without manual intervention.

Book Metric

Book Product List Schema

This table is a list of each Book Product. It is primarily used for drop-down fields, or where a list of all the books independent of metrics is desired.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
ProductForm	STRING	NULLABLE	The product form of the book
usage_flag	BOOLEAN	NULLABLE	Was there any usage detected, from any source, for this book
EditionNumber	INTEGER	NULLABLE	The edition number of the book
published_year	INTEGER	NULLABLE	The year the book was published
published_date	DATE	NULLABLE	The date the book was published
title	STRING	NULLABLE	The Books Title
bic_subjects	STRING	REPEATED	A list of BIC subjects
bisac_subjects	STRING	REPEATED	A list of BISAC subjects
thema_subjects	STRING	REPEATED	A list of thema subjects
keywords	STRING	REPEATED	A list of keywords
authors	RECORD	REPEATED	A list of Book Authors
authors.PersonName	STRING	NULLABLE	The author's full name in the format '[first name] [last name]'
authors.PersonNameInverted	STRING	NULLABLE	The author's full name in the format '[last name], [first name]'
authors.ORCID	STRING	NULLABLE	The Authors ORCID ID

Book Product Metrics Schema

This table contains metrics, organised by month, that are linked to each book. The country, city, institution, events and referrals expand on this to provided further useful breakdowns of metrics.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
title	STRING	NULLABLE	The title of the book
authors	RECORD	REPEATED	A list of Book Authors
authors.PersonName	STRING	NULLABLE	The author's full name in the format '[first name] [last name]'
authors.PersonNameInverted	STRING	NULLABLE	The author's full name in the format '[last name], [first name]'
authors.ORCID	STRING	NULLABLE	The Authors ORCID ID
published_year	INTEGER	NULLABLE	The Books published year

continues on next page

Table 22 – continued from previous page

name	type	mode	description
month	DATE	NULLABLE	The month in which the metrics took place
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref Events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	Quantity of sales
google_books_sales.countries	RECORD	REPEATED	A list of Countries
google_books_sales.countries.Country_of_Sale	STRING	NULLABLE	Country in which sale occured
google_books_sales.countries.qty	INTEGER	NULLABLE	Quantity of sales
jstor	RECORD	NULLABLE	Metrics from JSTOR

continues on next page

Table 22 – continued from previous page

name	type	mode	description
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of item requests
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.version	STRING	NULLABLE	Version of the OAPEN IRUS UK API, corresponds to the COUNTER report version
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.
ucl_discovery	RECORD	NULLABLE	Metrics from UCL Discovery
ucl_discovery.total_downloads	INTEGER	NULLABLE	
worldreader	RECORD	NULLABLE	Metrics from Worldreader
worldreader.total_downloads	INTEGER	NULLABLE	
internet_archive	RECORD	NULLABLE	Metrics from Internet Archive
internet_archive.total_downloads	INTEGER	NULLABLE	

Book Product Author Metrics Schema

This table contains metrics, organised by month and author, that are linked to each author.

Link to Query

name	type	mode	description
PersonName	STRING	NULLABLE	The author's full name in the format '[first name] [last name]'
PersonNameInverted	STRING	NULLABLE	The author's full name in the format '[last name], [first name]'
orcid	STRING	NULLABLE	Author's ORCID ID
unique_books	INTEGER	NULLABLE	Number of unique Books matched to the author

continues on next page

Table 23 – continued from previous page

name	type	mode	description
month	DATE	NULLABLE	Month in which metrics took place
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.views_total_country	INTEGER	NULLABLE	Number of page views aggregated over all countries
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	Number of sales
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	The total number of item requests
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform

continues on next page

Table 23 – continued from previous page

name	type	mode	description
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Year Metrics Schema

This table contains metrics, organised by published year and month, that are linked to each book.

Link to Query

name	type	mode	description
published_year	INTEGER	NULLABLE	Published Year
unique_books	INTEGER	NULLABLE	The number of unique books published that year in the dataset
month	DATE	NULLABLE	The month for which the metrics apply to
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads

continues on next page

Table 24 – continued from previous page

name	type	mode	description
crossref_events	RECORD	NULLABLE	Metrics from Crossref Events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.

continues on next page

Table 24 – continued from previous page

name	type	mode	description
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Event Metrics Schema

This table contains metrics, organised by month and crossref event type, that are linked to each book.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
title	STRING	NULLABLE	The title of the book
published_year	INTEGER	NULLABLE	The publisher year of the book
month	DATE	NULLABLE	The month for which the metrics apply to
event_source	STRING	NULLABLE	Event Source
crossref_events	RECORD	NULLABLE	Metrics from Crossref Events
crossref_events.count	INTEGER	NULLABLE	Count of Events

Book Product Metrics City Schema

This table contains metrics, organised by month and city of measured usage, that are linked to each book.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
title	STRING	NULLABLE	The title of the book
published_year	INTEGER	NULLABLE	The publisher year of the book
month	DATE	NULLABLE	The month for which the metrics apply to
city	STRING	NULLABLE	The name of the city
coordinates	STRING	NULLABLE	Geographical coordinates of the city
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01

continues on next page

Table 26 – continued from previous page

name	type	mode	description
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01

Book Product Metrics Country Schema

This table contains metrics, organised by month and country of measured usage, that are linked to each book.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
title	STRING	NULLABLE	The title of the book
published_year	INTEGER	NULLABLE	The publisher year of the book
month	DATE	NULLABLE	The month for which the metrics apply to
country_code	STRING	NULLABLE	The Country Code
country_name	STRING	NULLABLE	The Country Name
country_iso_name	STRING	NULLABLE	The ISO3166 (alpha 2) Country Name
country_wikipedia_name	STRING	NULLABLE	The Country Wikipedia Name
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads

continues on next page

Table 27 – continued from previous page

name	type	mode	description
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
ucl_discovery	RECORD	NULLABLE	Metrics from UCL Discovery
ucl_discovery.download_count	INTEGER	NULLABLE	Number of downloads
worldreader	RECORD	NULLABLE	Metrics from Worldreader
worldreader.download_count	INTEGER	NULLABLE	Number of downloads

Book Product Metrics Events Schema

This table contains metrics, organised by month and crossref event type, that are linked to each book.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
work_id	STRING	NULLABLE	Book Work ID
work_family_id	STRING	NULLABLE	Book Work Family ID
title	STRING	NULLABLE	The title of the book
published_year	INTEGER	NULLABLE	The publisher year of the book
month	DATE	NULLABLE	The month for which the metrics apply to
event_source	STRING	NULLABLE	Event Source
crossref_events	RECORD	NULLABLE	Metrics from Crossref Events
crossref_events.count	INTEGER	NULLABLE	Count of Events

Institution List Schema

This table is a list of each unique Institution where metrics are linked too. It is primarily used for drop-down fields, or where a list of all the institutions independent of metrics is desired.

Link to Query

name	type	mode	description
institution	STRING	NULLABLE	Institution Name

Book Product Metrics Institutions Schema

This table contains metrics, organised by month and institution for which there is measured activity linked to each book.

Link to Query

name	type	mode	description
product_id	STRING	NULLABLE	Book Product ID
title	STRING	NULLABLE	The title of the book
published_year	INTEGER	NULLABLE	The publisher year of the book
month	DATE	NULLABLE	The month for which the metrics apply to
institution	STRING	NULLABLE	Institution Name
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made from that specific institution

Book Product Metrics Publisher Schema

This index contains a summary of metrics, organised by month that are linked to each publisher.

Link to Query

name	type	mode	description
month	DATE	NULLABLE	Month for which these metrics apply
unique_books	INTEGER	NULLABLE	The number of unique books
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref Events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic

continues on next page

Table 31 – continued from previous page

name	type	mode	description
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	Quantity of sales
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of item requests
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Subjects BIC Schema

This table contains metrics, organised by month and BIC subject type, that are linked to each book.

Link to Query

name	type	mode	description
subject	STRING	NULLABLE	BIC Subject
subject_code	STRING	NULLABLE	BIC Subject Code
unique_books	INTEGER	NULLABLE	The number of unique books
month	DATE	NULLABLE	The month in which the metrics occurred
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)

continues on next page

Table 32 – continued from previous page

name	type	mode	description
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Subjects BISAC Schema

This table contains metrics, organised by month and BISAC subject type, that are linked to each book.

Link to Query

name	type	mode	description
subject	STRING	NULLABLE	BISAC Subject
subject_code	STRING	NULLABLE	BISAC Subject Code
unique_books	INTEGER	NULLABLE	The number of unique books
month	DATE	NULLABLE	The month in which the metrics occurred
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads

continues on next page

Table 33 – continued from previous page

name	type	mode	description
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01

continues on next page

Table 33 – continued from previous page

name	type	mode	description
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Subjects THEMA Schema

This table contains metrics, organised by month and THEMA subject type, that are linked to each book.

Link to Query

name	type	mode	description
subject	STRING	NULLABLE	Thema Subject
subject_code	STRING	NULLABLE	Thema Subject Code
unique_books	INTEGER	NULLABLE	The number of unique books
month	DATE	NULLABLE	The month in which the metrics occurred
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views

continues on next page

Table 34 – continued from previous page

name	type	mode	description
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

Book Product Subject Year Schema

This table contains metrics, organised by published year and month and currently just the BIC subject type, that are linked to each book.

Link to Query

name	type	mode	description
subject	STRING	NULLABLE	BIC subject (top level BIC subjects only)
published_year	INTEGER	NULLABLE	The published year of the book
unique_books	INTEGER	NULLABLE	The number of unique books
month	DATE	NULLABLE	The month in which the metrics occurred
google_analytics	RECORD	NULLABLE	Metrics from Google Analytics
google_analytics.page_views	INTEGER	NULLABLE	Number of page views
google_analytics.downloads	INTEGER	NULLABLE	Number of total downloads
google_analytics.downloads_pdf_book	INTEGER	NULLABLE	Number of PDF book downloads
google_analytics.downloads_pdf_chapter	INTEGER	NULLABLE	Number of PDF chapter downloads
google_analytics.downloads_html_chapter	INTEGER	NULLABLE	Number of HTML chapter downloads
google_analytics.downloads_epub_book	INTEGER	NULLABLE	Number of ePUB book downloads
google_analytics.downloads_epub_chapter	INTEGER	NULLABLE	Number of ePUB chapter downloads
google_analytics.downloads_mobi_book	INTEGER	NULLABLE	Number of MOBI book downloads
google_analytics.downloads_mobi_chapter	INTEGER	NULLABLE	Number of MOBI chapter downloads
crossref_events	RECORD	NULLABLE	Metrics from Crossref events
crossref_events.count	INTEGER	NULLABLE	Count of events
google_books_traffic	RECORD	NULLABLE	Metrics from Google Books Traffic
google_books_traffic.Book_Visits_BV_	INTEGER	NULLABLE	A Book Visit is registered each time a unique user views one of your books on Google Books. This number includes informational page views (such as the "About this book" page) as well as preview content page views
google_books_traffic.BV_with_Pages_Viewed	INTEGER	NULLABLE	The number of Book Visits in which users accessed preview pages of your book. This doesn't include visits where a user accessed only informational pages for your books
google_books_traffic.Non_Unique_Buy_Clicks	INTEGER	NULLABLE	The number of clicks on links for purchasing the book on retailer websites (including your website, if you provided a buy link)
google_books_traffic.BV_with_Buy_Clicks	INTEGER	NULLABLE	The number of visits which included a click on a purchase link
google_books_traffic.Buy_Link_CTR	FLOAT	NULLABLE	The clickthrough rate for purchase links. The values are percentages

continues on next page

Table 35 – continued from previous page

name	type	mode	description
google_books_traffic.Pages_Viewed	INTEGER	NULLABLE	The total number of unique preview content pages that a user viewed in a given session (counted as a 24-hour period)
google_books_sales	RECORD	NULLABLE	Metrics from Google Books Sales
google_books_sales.qty	INTEGER	NULLABLE	The number of units in the transaction. Negative for refunds
jstor	RECORD	NULLABLE	Metrics from JSTOR
jstor.Total_Item_Requests	INTEGER	NULLABLE	Total number of request made
irus_oapen	RECORD	NULLABLE	Metrics from OAPEN. Recorded with the IRUS-UK platform
irus_oapen.title_requests	INTEGER	NULLABLE	The total number of title requests. Only available for data before 2020-04-01
irus_oapen.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations. Only available for data since 2020-04-01
irus_oapen.total_item_requests	INTEGER	NULLABLE	The total number of item requests. Only available for data since 2020-04-01
irus_oapen.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations. Only available for data since 2020-04-01
irus_oapen.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests. Only available for data since 2020-04-01
irus_fulcrum	RECORD	NULLABLE	Metrics from Fulcrum. Recorded with the IRUS platform
irus_fulcrum.total_item_investigations	INTEGER	NULLABLE	The total number of item investigations.
irus_fulcrum.total_item_requests	INTEGER	NULLABLE	The total number of item requests.
irus_fulcrum.unique_item_investigations	INTEGER	NULLABLE	The number of unique item investigations.
irus_fulcrum.unique_item_requests	INTEGER	NULLABLE	The number of unique item requests.

QA Related Datasets

Unmatched Book Metrics Schema

This dataset is helpful for understanding where metrics and books defined in the onix feed are not matched. Helping target data quality tasks upstream of this workflow.

Link to Query

name	type	mode	description
ISBN	STRING	NULLABLE	The ISBN13
release_date	DATE	NULLABLE	The release date (month)
google_analytics_title	STRING	NULLABLE	The title of book, as specified in Google Analytics
google_books_title	STRING	NULLABLE	The title of book, as specified in Google Books
jstor_title	STRING	NULLABLE	The title of book, as specified in JSTOR

continues on next page

Table 36 – continued from previous page

name	type	mode	description
oapen_title	STRING	NULLABLE	The title of book, as specified in OAPEN/IRUS-UK
ucl_discovery_title	STRING	NULLABLE	The title of book, as specified in UCL Discovery
in_google_analytics	BOOLEAN	NULLABLE	Was this ISBN contained in Google Analytics
in_google_books	BOOLEAN	NULLABLE	Was this ISBN contained in Google Books
in_jstor	BOOLEAN	NULLABLE	Was this ISBN contained in JSTOR
in_oapen	BOOLEAN	NULLABLE	Was this ISBN contained in OAPEN
in_ucl_discovery	BOOLEAN	NULLABLE	Was this ISBN contained in UCL Discovery

3.1.3 License & contributing guidelines

Information about licenses, contributing guidelines etc.

License

```
                    Copyright 2019 Curtin University

                             Apache License
                        Version 2.0, January 2004
                     http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.
```

(continues on next page)

(continued from previous page)

```
    "Object" form shall mean any form resulting from mechanical
    transformation or translation of a Source form, including but
    not limited to compiled object code, generated documentation,
    and conversions to other media types.

    "Work" shall mean the work of authorship, whether in Source or
    Object form, made available under the License, as indicated by a
    copyright notice that is included in or attached to the work
    (an example is provided in the Appendix below).

    "Derivative Works" shall mean any work, whether in Source or Object
    form, that is based on (or derived from) the Work and for which the
    editorial revisions, annotations, elaborations, or other modifications
    represent, as a whole, an original work of authorship. For the purposes
    of this License, Derivative Works shall not include works that remain
    separable from, or merely link (or bind by name) to the interfaces of,
    the Work and Derivative Works thereof.

    "Contribution" shall mean any work of authorship, including
    the original version of the Work and any modifications or additions
    to that Work or Derivative Works thereof, that is intentionally
    submitted to Licensor for inclusion in the Work by the copyright owner
    or by an individual or Legal Entity authorized to submit on behalf of
    the copyright owner. For the purposes of this definition, "submitted"
    means any form of electronic, verbal, or written communication sent
    to the Licensor or its representatives, including but not limited to
    communication on electronic mailing lists, source code control systems,
    and issue tracking systems that are managed by, or on behalf of, the
    Licensor for the purpose of discussing and improving the Work, but
    excluding communication that is conspicuously marked or otherwise
    designated in writing by the copyright owner as "Not a Contribution."

    "Contributor" shall mean Licensor and any individual or Legal Entity
    on behalf of whom a Contribution has been received by Licensor and
    subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
    this License, each Contributor hereby grants to You a perpetual,
    worldwide, non-exclusive, no-charge, royalty-free, irrevocable
    copyright license to reproduce, prepare Derivative Works of,
    publicly display, publicly perform, sublicense, and distribute the
    Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
    this License, each Contributor hereby grants to You a perpetual,
    worldwide, non-exclusive, no-charge, royalty-free, irrevocable
    (except as stated in this section) patent license to make, have made,
    use, offer to sell, sell, import, and otherwise transfer the Work,
    where such license applies only to those patent claims licensable
    by such Contributor that are necessarily infringed by their
    Contribution(s) alone or by combination of their Contribution(s)
```

(continues on next page)

(continued from previous page)

```
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or
   for any such Derivative Works as a whole, provided Your use,
   reproduction, and distribution of the Work otherwise complies with
   the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
   any Contribution intentionally submitted for inclusion in the Work
   by You to the Licensor shall be under the terms and conditions of
```

(continues on next page)

(continued from previous page)

```
   this License, without any additional terms or conditions.
   Notwithstanding the above, nothing herein shall supersede or modify
   the terms of any separate license agreement you may have executed
   with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
   names, trademarks, service marks, or product names of the Licensor,
   except as required for reasonable and customary use in describing the
   origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
   agreed to in writing, Licensor provides the Work (and each
   Contributor provides its Contributions) on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
   implied, including, without limitation, any warranties or conditions
   of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
   PARTICULAR PURPOSE. You are solely responsible for determining the
   appropriateness of using or redistributing the Work and assume any
   risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
   whether in tort (including negligence), contract, or otherwise,
   unless required by applicable law (such as deliberate and grossly
   negligent acts) or agreed to in writing, shall any Contributor be
   liable to You for damages, including any direct, indirect, special,
   incidental, or consequential damages of any character arising as a
   result of this License or out of the use or inability to use the
   Work (including but not limited to damages for loss of goodwill,
   work stoppage, computer failure or malfunction, or any and all
   other commercial damages or losses), even if such Contributor
   has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing
   the Work or Derivative Works thereof, You may choose to offer,
   and charge a fee for, acceptance of support, warranty, indemnity,
   or other liability obligations and/or rights consistent with this
   License. However, in accepting such obligations, You may act only
   on Your own behalf and on Your sole responsibility, not on behalf
   of any other Contributor, and only if You agree to indemnify,
   defend, and hold each Contributor harmless for any liability
   incurred by, or claims asserted against, such Contributor by reason
   of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

   To apply the Apache License to your work, attach the following
   boilerplate notice, with the fields enclosed by brackets "[]"
   replaced with your own identifying information. (Don't include
   the brackets!)  The text should be enclosed in the appropriate
   comment syntax for the file format. We also recommend that a
```

(continues on next page)

(continued from previous page)

```
    file or class name and description of purpose be included on the
    same "printed page" as the copyright notice for easier
    identification within third-party archives.

Copyright 2019 Curtin University

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

Third Party Code

3.1.4 Python API reference

This page contains auto-generated API reference documentation[1].

oaebu_workflows

Subpackages

oaebu_workflows.dags

oaebu_workflows.database

Subpackages

oaebu_workflows.database.schema

oaebu_workflows.database.sql

oaebu_workflows.fixtures

oaebu_workflows.tests

Submodules

oaebu_workflows.tests.test_oaebu_partners

[1] Created with sphinx-autoapi

Module Contents

Classes

TestPartnerFromStr	A class whose instances are single test cases.

Attributes

MOCK_DATA_PARTNERS
MOCK_METADATA_PARTNERS

oaebu_workflows.tests.test_oaebu_partners.**MOCK_DATA_PARTNERS**

oaebu_workflows.tests.test_oaebu_partners.**MOCK_METADATA_PARTNERS**

class oaebu_workflows.tests.test_oaebu_partners.**TestPartnerFromStr**(*methodName='runTest'*)

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

> the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_valid_partner_name_string()

test_invalid_partner_name_string()

oaebu_workflows.tests.test_onix

Module Contents

Classes

TestOnixFunctions	Tests for the ONIX telescope

class oaebu_workflows.tests.test_onix.**TestOnixFunctions**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the ONIX telescope

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

test_onix_parser_download_execute()

Tests the onix_parser_download and onix_parser_execute functions

test_onix_collapse_subjects()

Tests the thoth_collapse_subjects function

test_onix_create_personname_fields()

Tests the function that creates the personname field

oaebu_workflows.workflows

Subpackages

oaebu_workflows.workflows.tests

Submodules

oaebu_workflows.workflows.tests.test_google_analytics3_telescope

Module Contents

Classes

TestGoogleAnalytics3Telescope	Tests for the Google Analytics telescope

Functions

create_http_mock_sequence(organisation_name)	Create a list of http mock sequences for listing books and getting dimension data

class oaebu_workflows.workflows.tests.test_google_analytics3_telescope.**TestGoogleAnalytics3Telescope**(**a*
**

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Google Analytics telescope

Constructor which sets up variables used by tests. :param args: arguments. :param kwargs: keyword arguments.

test_dag_structure()

Test that the Google Analytics DAG has the correct structure. :return: None

test_dag_load()

Test that the Google Analytics DAG can be loaded from a DAG bag.

test_telescope(*mock_account_credentials*, *mock_build*)

Test the Google Analytics telescope end to end specifically for ANU Press, to test custom dimensions. :return: None.

oaebu_workflows.workflows.tests.test_google_analytics3_telescope.**create_http_mock_sequence**(*organisation_na*

Create a list of http mock sequences for listing books and getting dimension data

Parameters
organisation_name (*str*) – The organisation name (add custom dimensions for ANU)

Returns
A list with HttpMockSequence instances

Return type
list

oaebu_workflows.workflows.tests.test_google_books_telescope

Module Contents

Classes

TestGoogleBooksTelescope	Tests for the GoogleBooks telescope

class oaebu_workflows.workflows.tests.test_google_books_telescope.**TestGoogleBooksTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the GoogleBooks telescope

Constructor which sets up variables used by tests. :param args: arguments. :param kwargs: keyword arguments.

test_dag_structure()

Test that the Google Books DAG has the correct structure.

test_dag_load()

Test that the Google Books DAG can be loaded from a DAG bag.

test_telescope()

Test the Google Books telescope end to end.

test_gb_transform(*mock_variable_get*)

Test sanity check in transform method when transaction date falls outside release month

Parameters

mock_variable_get – Mock Airflow Variable 'data'

oaebu_workflows.workflows.tests.test_irus_fulcrum_telescope

Module Contents

Classes

TestIrusFulcrumTelescope	Tests for the Fulcrum telescope

Attributes

FAKE_PUBLISHERS

oaebu_workflows.workflows.tests.test_irus_fulcrum_telescope.**FAKE_PUBLISHERS** = ['Fake Publisher 1', 'Fake Publisher 2', 'Fake Publisher 3']

class oaebu_workflows.workflows.tests.test_irus_fulcrum_telescope.**TestIrusFulcrumTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Fulcrum telescope

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

test_dag_structure()

Test that the ONIX DAG has the correct structure and raises errors when necessary

test_dag_load()

Test that the DAG can be loaded from a DAG bag.

test_telescope()

Test the Fulcrum telescope end to end.

test_download_fulcrum_month_data()

Tests the download_fuclrum_month_data function

test_transform_fulcrum_data()

Tests the transform_fulcrum_data function

oaebu_workflows.workflows.tests.test_irus_oapen_telescope

Module Contents

Classes

TestIrusOapenTelescope	Tests for the Oapen Irus Uk telescope

class oaebu_workflows.workflows.tests.test_irus_oapen_telescope.**TestIrusOapenTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Oapen Irus Uk telescope

Constructor which sets up variables used by tests. :param args: arguments. :param kwargs: keyword arguments.

test_dag_structure()

Test that the Oapen Irus Uk DAG has the correct structure.

test_dag_load()

Test that the Oapen Irus Uk DAG can be loaded from a DAG bag.

test_telescope(*mock_authorized_session*, *mock_account_credentials*, *mock_build*)

Test the Oapen Irus Uk telescope end to end.

test_create_cloud_function(*mock_create_function*, *mock_function_exists*, *mock_upload*, *mock_variable_get*)

Test the create_cloud_function method of the IrusOapenRelease

Parameters
mock_variable_get – Mock Airflow Variable 'data'

test_call_cloud_function(*mock_function_exists*, *mock_call_function*, *mock_conn_get*, *mock_variable_get*)

Test the call_cloud_function method of the IrusOapenRelease

Parameters
mock_variable_get – Mock Airflow Variable 'data'

test_upload_source_code_to_bucket(*mock_create_bucket*, *mock_upload_to_bucket*)

Test getting source code from oapen irus uk release and uploading to storage bucket. Test expected results both when md5 hashes match and when they don't.

test_cloud_function_exists()

Test the function that checks whether the cloud function exists

test_create_cloud_function()

Test the function that creates the cloud function

test_call_cloud_function(*mock_authorized_session*)

Test the function that calls the cloud function

oaebu_workflows.workflows.tests.test_jstor_telescope

Module Contents

Classes

TestJstorTelescope	Tests for the Jstor telescope

Functions

create_http_mock_sequence(country_report_url, ...)	Create a list with mocked http responses

class oaebu_workflows.workflows.tests.test_jstor_telescope.**TestJstorTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Jstor telescope

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

test_dag_structure()

Test that the Jstor DAG has the correct structure.

test_dag_load()

Test that the Jstor DAG can be loaded from a DAG bag.

test_telescope(*mock_account_credentials*, *mock_build*)

Test the Jstor telescope end to end.

test_get_label_id()

Test getting label id both when label already exists and does not exist yet.

test_get_release_date()

Test that the get_release_date returns the correct release date and raises an exception when dates are incorrect

oaebu_workflows.workflows.tests.test_jstor_telescope.**create_http_mock_sequence**(*country_report_url*, *institution_report_url*, *wrong_publisher_report_url*)

Create a list with mocked http responses

Parameters

- **country_report_url** (*str*) – URL to country report
- **institution_report_url** (*str*) – URL to institution report
- **wrong_publisher_report_url** (*str*) – URL to report with a non-matching publisher id

Returns
List with http responses

Return type
list

oaebu_workflows.workflows.tests.test_oapen_metadata_telescope

Module Contents

Classes

TestOapenMetadataTelescope	Tests for the Oapen Metadata Telescope DAG
TestDownloadMetadata	A class whose instances are single test cases.
TestFilterThroughSchema	A class whose instances are single test cases.
TestRemoveInvalidProducts	A class whose instances are single test cases.
TestFindOnixProduct	A class whose instances are single test cases.

class oaebu_workflows.workflows.tests.test_oapen_metadata_telescope.**TestOapenMetadataTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Oapen Metadata Telescope DAG

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

test_dag_structure()

Test that the Oapen Metadata DAG has the correct structure

test_dag_load()

Test that the OapenMetadata DAG can be loaded from a DAG bag

test_telescope()

Test telescope task execution.

class oaebu_workflows.workflows.tests.test_oapen_metadata_telescope.**TestDownloadMetadata**(*methodName='run*

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

> the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

valid_download_cassette

invalid_download_cassette

empty_download_cassette

bad_response_cassette

header_only_download_cassette

valid_download_xml

uri = 'https://library.oapen.org/download-export?format=onix'

test_download_metadata()

Test that metadata successfully downloads after 200 respose

test_download_metadata_invalid_xml()

Test behaviour when the downloaded file is an invalid XML

test_download_metadata_empty_xml()

Test behaviour when the downloaded file is an empty XML

test_download_metadata_no_products()

Test behaviour when the downloaded file is an empty XML

test_download_metadata_bad_response()

Test behaviour when the downloaded file has a non-200 response code

class oaebu_workflows.workflows.tests.test_oapen_metadata_telescope.TestFilterThroughSchema(*methodName=*

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_filter_through_schema()

Tests the generic use case of the function

test_matching_keys()

Tests that the function correctly processes the input dictionary when all nested keys match the schema

test_empty_input()

Tests that the function correctly handles an empty input dictionary

test_no_matching_keys()

Tests that the function correctly handles an input dictionary with no matching keys in the schema

test_edge_case_empty_schema()

Tests that the function correctly handles an empty schema

class oaebu_workflows.workflows.tests.test_oapen_metadata_telescope.TestRemoveInvalidProducts(*methodNam*

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

valid_parsed_xml

invalid_products_removed_xml

empty_xml

invalid_products_xml

test_remove_invalid_products()

Tests the function used to remove invalid products from an xml file

test_empty_xml()

Tests the function used to remove invalid products from an xml file

class oaebu_workflows.workflows.tests.test_oapen_metadata_telescope.**TestFindOnixProduct**(*methodName='runT*

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

`valid_input = ['<ONIXMessage>', '<Product>', '<RecordReference>1</RecordReference>', '</Product>',...`

`test_find_onix_product()`

Test that the function can extract multiple products from a valid input xml

`test_out_of_bounds_supplied()`

Test that errors are thrown when improper input is supplied

`test_missing_record_reference()`

Tests that a product without a RecordReference raises a KeyError

`test_empty_product()`

Tests that a product without a RecordReference raises a KeyError

`test_no_product_tags()`

Tests that the function raises a ValueError when <Product> tags are not closed or missing

oaebu_workflows.workflows.tests.test_onix_telescope

Module Contents

Classes

TestOnixTelescope	Tests for the ONIX telescope

`class oaebu_workflows.workflows.tests.test_onix_telescope.TestOnixTelescope(*args, **kwargs)`

Bases: `observatory.platform.observatory_environment.ObservatoryTestCase`

Tests for the ONIX telescope

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

`test_dag_structure()`

Test that the ONIX DAG has the correct structure.

test_dag_load()

Test that the Geonames DAG can be loaded from a DAG bag.

test_telescope()

Test the ONIX telescope end to end.

oaebu_workflows.workflows.tests.test_onix_work_aggregation

Module Contents

Classes

TestUnionFind	Test the UnionFind class.
TestBookWork	Test the BookWork class.
TestBookWorkFamily	Test the BookWorkFamily class.
TestGetPrefProductId	Test the get_pref_product_id function.
TestGetPrefWorkId	Test the get_pref_work_id function.
TestBookWorkAggregator	Test the BookWorkAggregator class.
TestBookWorkFamilyAggregator	Test the BookWorkFamilyAggregator class.

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestUnionFind**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the UnionFind class.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_root_trivial()

test_unite_two()

test_unite_two_parts()

test_unite_two_parts_merge()

test_find_trivial()

test_find_after_merge()

test_find_after_two_parts_merge()

test_get_partition_trivial()

test_get_partition_two_parts()

test_get_partition_two_parts_merge()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestBookWork**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the BookWork class.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_ctor()

test_add_product()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestBookWorkFamily**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the BookWorkFamily class.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_ctor()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestGetPrefProductId**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the get_pref_product_id function.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_unknown()

test_get_pidproprietary()

test_get_doi()

test_get_gtin13()

test_get_isbn()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestGetPrefWorkId**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the get_pref_work_id function.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_unknown()

test_pid_proprietary()

test_doi()

test_gtin13()

test_isbn13()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestBookWorkAggregator**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the BookWorkAggregator class.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_ctor()

test_is_relevant_worK_relation()

test_aggregate_empty()

test_agg_relworks()

test_agg_relworks_gtin13()

test_agg_relworks_pid_proprietary()

test_agg_products()

test_agg_relprod_doi()

test_agg_relprod_gtin13()

test_aggregate1()

test_aggregate2()

test_aggregate3()

test_agg_works_products_composite()

test_get_pid_idx_missing()

test_get_pid_idx_unknown()

test_agg_relprod_missing_record()

test_get_works_lookup_table()

log_agg_related_product_errors()

test_log_agg_relworks_errors()

test_log_agg_relworks_errors_miss_gtin()

test_log_get_works_lookup_table_errors()

test_filtering_duplicate_isbns()

class oaebu_workflows.workflows.tests.test_onix_work_aggregation.**TestBookWorkFamilyAggregator**(**args*, ***kwargs*)

Bases: unittest.TestCase

Test the BookWorkFamilyAggregator class.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_ctor()

test_agg_relproducts()

test_aggregate_products_missing()

test_aggregate_products()

test_aggregate_products_gtin13()

test_aggregate_products_pid_proprietary()

test_get_wid_unsupported()

test_get_wid_idx_unsupported()

test_get_wid_idx_missing_isbn()

test_get_wid_idx_missing_gtin()

test_get_wid_idx_missing_pid_proprietary()

test_get_works_family_lookup_table()

oaebu_workflows.workflows.tests.test_onix_workflow

Module Contents

Classes

TestOnixWorkflow	Functionally test the workflow

class oaebu_workflows.workflows.tests.test_onix_workflow.**TestOnixWorkflow**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Functionally test the workflow

onix_data

test_make_release(*mock_sel_table_suffixes*)

Tests that the make_release function works as intended

test_cleanup()

Tests the cleanup function of the workflow

test_dag_load()

Test that the DAG loads

test_dag_structure()

Tests that the dag structure is created as expected on dag load

test_create_and_load_aggregate_works_table(*mock_bq_query*)

test_crossref_API_calls()

Test the functions that query the crossref event and metadata APIs

test_crossref_transform()

Test the function that transforms the crossref events data

test_utility_functions()

Test the standalone functions in the Onix workflow that aren't specifially tested in other classes

setup_fake_lookup_tables(*settings_dataset_id*, *fixtures_dataset_id*, *release_date*, *bucket_name*)

Create a new onix and subject lookup and country tables with their own dataset and table ids. Populate them with some fake data.

Parameters

- **settings_dataset_id** (*str*) – The dataset to store the country table
- **fixtures_dataset_id** (*str*) – The dataset to store the lookup tables
- **release_date** (*pendulum.DateTime*) – The release/snapshot date
- **bucket_name** (*str*) – The GCP bucket name to load the data to

setup_input_data(*settings_dataset_id*, *fixtures_dataset_id*, *crossref_master_dataset_id*, *partner_dataset*, *onix_dataset_id*, *release_date*, *bucket_name*, *include_google_analytics3*)

Uploads the data partner fixtures to their respective GCP bucket and bigquery tables. Creates the partners based on the originals - but changes the dataset ids for tests. Create a new onix and subject lookup and country tables with their own dataset and table ids.

Parameters

- **settings_dataset_id** (*str*) – The dataset to store the country table
- **fixtures_dataset_id** (*str*) – The dataset to store the lookup tables
- **partner_dataset** (*str*) – The bigquery dataset ID to load the data partner tables to
- **crossref_master_dataset_id** (*str*) – The bigquery dataset ID of the master crossref table
- **onix_dataaset** – The Bigquery dataset ID to load the onix partner table to
- **release_date** (*pendulum.DateTime*) – The release/snapshot date of sharded tables
- **bucket_name** (*str*) – The name of the bucket to upload the jsonl files to
- **include_google_analytics3** (*bool*) – Whether to include google analytics 3 as a partner
- **onix_dataset_id** (*str*) –

Returns

The resulting OaebuPartners

Return type

List[*oaebu_workflows.oaebu_partners.OaebuPartner*]

run_telescope_tests(*, *include_google_analytics3=False*)

Functional test of the ONIX workflow

Parameters

include_google_analytics3 (*bool*) –

test_telescope()

Test that ONIX Workflow runs when Google Analytics is not included

test_telescope_with_google_analytics()

Test that ONIX Workflow runs when Google Analytics is included

test_get_onix_records(*mock_bq_query*)

oaebu_workflows.workflows.tests.test_thoth_telescope

Module Contents

Classes

TestThothTelescope	Tests for the Thoth telescope

Attributes

FAKE_PUBLISHER_ID

oaebu_workflows.workflows.tests.test_thoth_telescope.**FAKE_PUBLISHER_ID** = **'fake_publisher_id'**

class oaebu_workflows.workflows.tests.test_thoth_telescope.**TestThothTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Thoth telescope

Constructor which sets up variables used by tests.

Parameters

- **args** – arguments.
- **kwargs** – keyword arguments.

test_dag_structure()

Test that the ONIX DAG has the correct structure.

test_dag_load()

Test that the DAG can be loaded from a DAG bag.

test_telescope()

Test the Thoth telescope end to end.

test_download_onix()

Tests the download_onix function. Will test that the function works as expected using proxy HTTP requests

test_thoth_api()

Tests that HTTP requests to the thoth API are successful

oaebu_workflows.workflows.tests.test_ucl_discovery_telescope

Module Contents

Classes

TestUclDiscoveryTelescope	Tests for the Ucl Discovery telescope
TestGetIsbnEprintMappings	Tests for the get_isbn_eprint_mappings function
TestDownloadDiscoveryStats	Tests for the download_discovery_stats function
TestTransformDiscoveryStats	A class whose instances are single test cases.

class oaebu_workflows.workflows.tests.test_ucl_discovery_telescope.**TestUclDiscoveryTelescope**(**args*, ***kwargs*)

Bases: observatory.platform.observatory_environment.ObservatoryTestCase

Tests for the Ucl Discovery telescope

Constructor which sets up variables used by tests.

test_dag_structure()

Test that the UCL Discovery DAG has the correct structure.

test_dag_load()

Test that the UCL Discovery DAG can be loaded from a DAG bag.

test_telescope()

Test the UCL Discovery telescope end to end.

class oaebu_workflows.workflows.tests.test_ucl_discovery_telescope.**TestGetIsbnEprintMappings**(**args*, ***kwargs*)

Bases: unittest.TestCase

Tests for the get_isbn_eprint_mappings function

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_get_isbn_eprint_mappings(*mock_build*, *mock_get_connection*, *mock_sa*)

test_invalid_header(*mock_build*, *mock_get_connection*, *mock_sa*)

test_empty_sheet(*mock_build*, *mock_get_connection*, *mock_sa*)

test_missing_values(*mock_build*, *mock_get_connection*, *mock_sa*)

class oaebu_workflows.workflows.tests.test_ucl_discovery_telescope.**TestDownloadDiscoveryStats**(**args*, ***kwargs*)

Bases: unittest.TestCase

Tests for the download_discovery_stats function

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_download_discovery_stats(*mock_retry_get_url*)

Test the download_discovery_stats function works with correct inputs

test_download_discovery_stats_invalid_timescale(*mock_retry_get_url*)

Check if exceptions raised when timescale is inconsistent with inputs

test_download_discovery_stats_invalid_eprint_id(*mock_retry_get_url*)

Check if exceptions raised when eprint ID is inconsistent with inputs

class oaebu_workflows.workflows.tests.test_ucl_discovery_telescope.**TestTransformDiscoveryStats**(*methodNar*

Bases: unittest.TestCase

A class whose instances are single test cases.

By default, the test code itself should be placed in a method named 'runTest'.

If the fixture may be used for many test cases, create as many test methods as are needed. When instantiating such a TestCase subclass, specify in the constructor arguments the name of the test method that the instance is to execute.

Test authors should subclass TestCase for their own tests. Construction and deconstruction of the test's environment ('fixture') can be implemented by overriding the 'setUp' and 'tearDown' methods respectively.

If it is necessary to override the __init__ method, the base class __init__ method must always be called. It is important that subclasses should not change the signature of their __init__ method, since instances of the classes are instantiated automatically by parts of the framework in order to be run.

When subclassing TestCase, you can set these attributes: * failureException: determines which exception will be raised when

the instance's assertion methods fail; test methods raising this exception will be deemed to have 'failed' rather than 'errored'.

- **longMessage: determines whether long messages (including repr of**
  objects used in assert methods) will be printed on failure in *addition* to any explicit message passed.
- **maxDiff: sets the maximum length of a diff in failure messages**
  by assert methods using difflib. It is looked up as an instance attribute so can be configured by individual tests if required.

Create an instance of the class that will use the named test method when executed. Raises a ValueError if the instance does not have a method with the specified name.

test_transform_discovery_stats()

Test the transform_discovery_stats function when inputs are valid

test_transform_discovery_stats_no_country_records()

Test the transform_discovery_stats function when country records are missing

test_transform_discovery_stats_mismatching_eprint_ids()

Test the transform_discovery_stats function when eprint IDs do not match

test_transform_discovery_stats_mismatching_timescales()

Test the transform_discovery_stats function when timescales do not match

Submodules

oaebu_workflows.workflows.google_analytics3_telescope

Module Contents

Classes

GoogleAnalytics3Release	Construct a GoogleAnalytics3Release.
GoogleAnalytics3Telescope	Google Analytics Telescope.

Functions

initialize_analyticsreporting(...)	Initializes an Analytics Reporting API V4 service object.
list_all_books(service, view_id, pagepath_regex, ...)	List all available books by getting all pagepaths of a view id in a given period.
create_book_result_dicts(book_entries, ...)	Create a dictionary to store results for a single book. Pagepath, title and avg time on page are already given.
get_dimension_data(service, view_id, ...)	Get reports data from the Google Analytics Reporting service for a single dimension and multiple metrics.
add_to_book_result_dict(book_results, dimension, ...)	Add the 'unique_views', 'page_views' and 'sessions' results to the book results dict if these metrics are of interest for the
get_reports(service, organisation_name, view_id, ...)	Get reports data from the Google Analytics Reporting API.

class oaebu_workflows.workflows.google_analytics3_telescope.**GoogleAnalytics3Release**(*dag_id*, *run_id*, *data_interval_start*, *data_interval_end*, *partition_date*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Construct a GoogleAnalytics3Release.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID
- **data_interval_start** (*pendulum.DateTime*) – The start date of the DAG the start date of the download period.
- **data_interval_end** (*pendulum.DateTime*) – end date of the download period, also used as release date for BigQuery table and file paths
- **partition_date** (*pendulum.DateTime*) –

class oaebu_workflows.workflows.google_analytics3_telescope.**GoogleAnalytics3Telescope**(*dag_id*, *organisation_name*, *cloud_workspace*, *view_id*, *pagepath_regex*, *data_partner='google_* *bq_dataset_description* *from* *Google* *sources'*, *bq_table_description=* *api_dataset_id='googl* *oaebu_service_accoun* *observatory_api_conn_* *catchup=True*, *start_date=pendulum.d* *1,* *1)*, *schedule='@monthly'*)

Bases: observatory.platform.workflows.workflow.Workflow

Google Analytics Telescope.

Construct a GoogleAnalytics3Telescope instance. :param dag_id: The ID of the DAG :param organisation_name: The organisation name as per Google Analytics :param cloud_workspace: The CloudWorkspace object for this DAG :param view_id: The Google Analytics view ID :param pagepath_regex: The pagepath regex :param data_partner: The name of the data partner :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param oaebu_service_account_conn_id: Airflow connection ID for the OAEBU service account :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param catchup: Whether to catchup the DAG or not :param start_date: The start date of the DAG :param schedule: The schedule interval of the DAG

Parameters

- **dag_id** (*str*) –
- **organisation_name** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **view_id** (*str*) –
- **pagepath_regex** (*str*) –
- **data_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **oaebu_service_account_conn_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **start_date** (*pendulum.DateTime*) –

- **schedule** (*str*) –

ANU_ORG_NAME = 'ANU Press'

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'.

Parameters
kwargs – the context passed from the PythonOperator.

Return type
List[*GoogleAnalytics3Release*]

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: A list of grid release instances

check_dependencies(***kwargs*)

Check dependencies of DAG. Add to parent method to additionally check for a view id and pagepath regex

Parameters
kwargs – the context passed from the Airflow Operator.

Returns
True if dependencies are valid.

Return type
bool

download_transform(*releases*, ***kwargs*)

Task to download and transform the google analytics release for a given month.

Parameters
releases (*List[GoogleAnalytics3Release]*) – a list with one google analytics release.

Return type
None

upload_transformed(*releases*, ***kwargs*)

Uploads the transformed file to GCS

Parameters
releases (*List[GoogleAnalytics3Release]*) –

Return type
None

bq_load(*releases*, ***kwargs*)

Loads the data into BigQuery

Parameters
releases (*List[GoogleAnalytics3Release]*) –

Return type
None

add_new_dataset_releases(*releases*, ***kwargs*)

Adds release information to API.

Parameters
releases (*List[GoogleAnalytics3Release]*) –

Return type
None

cleanup(*releases*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
releases (*List[GoogleAnalytics3Release]*) –

Return type
None

oaebu_workflows.workflows.google_analytics3_telescope.**initialize_analyticsreporting**(*oaebu_service_account_*

Initializes an Analytics Reporting API V4 service object.

Returns
An authorized Analytics Reporting API V4 service object.

Parameters
oaebu_service_account_conn_id (*str*) –

Return type
googleapiclient.discovery.Resource

oaebu_workflows.workflows.google_analytics3_telescope.**list_all_books**(*service*, *view_id*, *pagepath_regex*, *data_interval_start*, *data_interval_end*, *organisation_name*, *metrics*)

List all available books by getting all pagepaths of a view id in a given period. Note: Google API will not return a result for any entry in which all supplied metrics are zero. However, it will return 'some' results if you supply no metrics, contrary to the documentation. Date ranges are inclusive.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – The Google Analytics Reporting service object.
- **view_id** (*str*) – The view id.
- **pagepath_regex** (*str*) – The regex expression for the pagepath of a book.
- **data_interval_start** (*pendulum.DateTime*) – The start date of the DAG Start date of analytics period
- **data_interval_end** (*pendulum.DateTime*) – End date of analytics period
- **organisation_name** (*str*) – The organisation name.
- **metrics** (*list*) –

Param
metrics: The metrics to return return with the book results

Returns
A list with dictionaries, one for each book entry (the dict contains the pagepath, title and average time

Return type
Tuple[List[dict], list]

on page) and a list of all pagepaths.

oaebu_workflows.workflows.google_analytics3_telescope.**create_book_result_dicts**(*book_entries*, *data_interval_start*, *data_interval_end*, *organisation_name*)

Create a dictionary to store results for a single book. Pagepath, title and avg time on page are already given. The other metrics will be added to the dictionary later.

Parameters

- **book_entries** (*List[dict]*) – List with dictionaries of book entries.
- **data_interval_start** (*pendulum.DateTime*) – The start date of the DAG Start date of analytics period.
- **data_interval_end** (*pendulum.DateTime*) – End date of analytics period.
- **organisation_name** (*str*) – The organisation name.

Returns
Dict to store results

Return type
Dict[dict]

oaebu_workflows.workflows.google_analytics3_telescope.**get_dimension_data**(*service*, *view_id*, *data_interval_start*, *data_interval_end*, *metrics*, *dimension*, *pagepaths*)

Get reports data from the Google Analytics Reporting service for a single dimension and multiple metrics. The results are filtered by pagepaths of interest and ordered by pagepath as well.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – The Google Analytics Reporting service.
- **view_id** (*str*) – The view id.
- **data_interval_start** (*pendulum.DateTime*) – The start date of the DAG The start date of the analytics period.
- **data_interval_end** (*pendulum.DateTime*) – The end date of the analytics period.
- **metrics** (*list*) – List with dictionaries of metric.
- **dimension** (*dict*) – The dimension.
- **pagepaths** (*list*) – List with pagepaths to filter and sort on.

Returns
List with reports data for dimension and metrics.

Return type
list

oaebu_workflows.workflows.google_analytics3_telescope.**add_to_book_result_dict**(*book_results*, *dimension*, *pagepath*, *unique_views*, *page_views*, *sessions*)

Add the 'unique_views', 'page_views' and 'sessions' results to the book results dict if these metrics are of interest for the current dimension.

Parameters

- **book_results** (*dict*) – A dictionary with all book results.
- **dimension** (*dict*) – Current dimension for which 'unique_views' and 'sessions' data is given.
- **pagepath** (*str*) – Pagepath of the book.
- **unique_views** (*dict*) – Number of unique views for the pagepath&dimension
- **page_views** (*dict*) – Number of page views for the pagepath&dimension
- **sessions** (*dict*) – Number of sessions for the pagepath&dimension

Returns
None

oaebu_workflows.workflows.google_analytics3_telescope.**get_reports**(*service*, *organisation_name*, *view_id*, *pagepath_regex*, *data_interval_start*, *data_interval_end*)

Get reports data from the Google Analytics Reporting API.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – The Google Analytics Reporting service.
- **organisation_name** (*str*) – Name of the organisation.
- **view_id** (*str*) – The view id.
- **pagepath_regex** (*str*) – The regex expression for the pagepath of a book.
- **data_interval_start** (*pendulum.DateTime*) – The start date of the DAG Start date of analytics period
- **data_interval_end** (*pendulum.DateTime*) – End date of analytics period

Returns
List with google analytics data for each book

Return type
list

oaebu_workflows.workflows.google_books_telescope

Module Contents

Classes

GoogleBooksRelease	Construct a GoogleBooksRelease.
GoogleBooksTelescope	The Google Books telescope.

Functions

gb_transform(download_files, sales_path, traffic_path, ...)	Transforms sales and traffic reports. For both reports it transforms the csv into a jsonl file and

class oaebu_workflows.workflows.google_books_telescope.**GoogleBooksRelease**(*dag_id*, *run_id*, *partition_date*, *sftp_files*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Construct a GoogleBooksRelease.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID
- **partition_date** (*pendulum.DateTime*) – the partition date, corresponds to the last day of the month being processed.
- **sftp_files** (*List[str]*) – List of full filepaths to download from sftp service (incl. in_progress folder)

class oaebu_workflows.workflows.google_books_telescope.**GoogleBooksTelescope**(*dag_id*, *cloud_workspace*, *sftp_root='/'*, *sales_partner='google_books_sales*, *traffic_partner='google_books_traf*, *bq_dataset_description='Data from Google sources'*, *bq_sales_table_description=None*, *bq_traffic_table_description=None*, *api_dataset_id='google_books'*, *sftp_service_conn_id='sftp_service*, *observatory_api_conn_id=AirflowC*, *catchup=False*, *schedule='@weekly'*, *start_date=pendulum.datetime(201*, *1, 1)*)

Bases: observatory.platform.workflows.workflow.Workflow

The Google Books telescope.

Construct a GoogleBooksTelescope instance. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param sftp_root: The root of the SFTP filesystem to work with :param sales_partner: The name of the sales partner :param traffic_partner: The name of the traffic partner :param bq_dataset_description: Description for the BigQuery dataset :param bq_sales_table_description: Description for the BigQuery Google Books Sales table :param bq_traffic_table_description: Description for the BigQuery Google Books Traffic table :param api_dataset_id: The ID to store the dataset release in the API :param sftp_service_conn_id: Airflow connection ID for the SFTP service :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param catchup: Whether to catchup the DAG or not :param schedule: The schedule interval of the DAG :param start_date: The start date of the DAG

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **sftp_root** (*str*) –
- **sales_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **traffic_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_sales_table_description** (*str*) –
- **bq_traffic_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **sftp_service_conn_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **schedule** (*str*) –
- **start_date** (*pendulum.DateTime*) –

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'.

Parameters
kwargs – the context passed from the PythonOperator.

Return type
List[*GoogleBooksRelease*]

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: A list of google books release instances

list_release_info(***kwargs*)

Lists all Google Books releases available on the SFTP server and publishes sftp file paths and release_date's as an XCom.

Returns
the identifier of the task to execute next.

Return type
bool

move_files_to_in_progress(*releases*, ***kwargs*)

Move Google Books files to SFTP in-progress folder.

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

download(*releases*, ***kwargs*)

Task to download the Google Books releases for a given month.

Parameters
releases (*List[GoogleBooksRelease]*) –

upload_downloaded(*releases*, ***kwargs*)

Uploads the downloaded files to GCS for each release

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

transform(*releases*, ***kwargs*)

Task to transform the Google Books releases for a given month.

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

upload_transformed(*releases*, ***kwargs*)

Uploads the transformed files to GCS for each release

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

move_files_to_finished(*releases*, ***kwargs*)

Move Google Books files to SFTP finished folder.

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

bq_load(*releases*, ***kwargs*)

Loads the sales and traffic data into BigQuery

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

add_new_dataset_releases(*releases*, ***kwargs*)

Adds release information to API.

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

cleanup(*releases*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
releases (*List[GoogleBooksRelease]*) –

Return type
None

oaebu_workflows.workflows.google_books_telescope.**gb_transform**(*download_files*, *sales_path*, *traffic_path*, *release_date*)

Transforms sales and traffic reports. For both reports it transforms the csv into a jsonl file and replaces spaces in the keys with underscores.

Parameters

- **download_files** (*Tuple[str, str]*) – The Google Books Sales and Traffic files
- **sales_path** (*str*) – The file path to save the transformed sales data to
- **traffic_path** (*str*) – The file path to save the transformed traffic data to
- **release_date** (*pendulum.DateTime*) – The release date to use as a partitioning date

Return type
None

oaebu_workflows.workflows.irus_fulcrum_telescope

Module Contents

Classes

IrusFulcrumRelease	Create a IrusFulcrumRelease instance.
IrusFulcrumTelescope	The Fulcrum Telescope

Functions

download_fulcrum_month_data(download_month, requestor_id)	Download Fulcrum data for the release month
transform_fulcrum_data(totals_data, country_data[, ...])	Transforms Fulcrum downloaded "totals" and "country" data.

Attributes

IRUS_FULCRUM_ENDPOINT_TEMPLATE

oaebu_workflows.workflows.irus_fulcrum_telescope.**IRUS_FULCRUM_ENDPOINT_TEMPLATE** = **'https://irus.jisc.ac.uk/api/v3/irus/reports/irus_ir/?platform=235&requestor_id={requestor_id}&beg...'**

class oaebu_workflows.workflows.irus_fulcrum_telescope.**IrusFulcrumRelease**(*dag_id*, *run_id*, *data_interval_start*, *data_interval_end*, *partition_date*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Create a IrusFulcrumRelease instance.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The airflow run ID
- **data_interval_start** (*pendulum.DateTime*) – The beginning of the data interval
- **data_interval_end** (*pendulum.DateTime*) – The end of the data interval
- **partition_date** (*pendulum.DateTime*) – The release/partition date

class oaebu_workflows.workflows.irus_fulcrum_telescope.**IrusFulcrumTelescope**(*dag_id*, *cloud_workspace*, *publishers*, *data_partner='irus_fulcrum'*, *bq_dataset_description='IRUS dataset'*, *bq_table_description=None*, *api_dataset_id='fulcrum'*, *observatory_api_conn_id=AirflowC irus_oapen_api_conn_id='irus_api catchup=True*, *schedule='0 0 4 * *'*, *start_date=pendulum.datetime(202 4, 1)*)

Bases: observatory.platform.workflows.workflow.Workflow

The Fulcrum Telescope :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param publishers: The publishers pertaining to this DAG instance (as listed in Fulcrum) :param data_partner: The name of the data partner :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param irus_oapen_api_conn_id: Airflow connection ID OAPEN IRUS UK (counter 5) :param catchup: Whether to catchup the DAG or not :param schedule: The schedule interval of the DAG :param start_date: The start date of the DAG

Parameters

- **dag_id** (*str*) –

- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **publishers** (*List[str]*) –
- **data_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **irus_oapen_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **schedule** (*str*) –
- **start_date** (*pendulum.DateTime*) –

make_release(***kwargs*)

Create a IrusFulcrumRelease instance Dates are best explained with an example Say the dag is scheduled to run on 2022-04-07 Interval_start will be 2022-03-01 Interval_end will be 2022-04-01 partition_date will be 2022-03-31

Return type
IrusFulcrumRelease

download(*release*, ***kwargs*)

Task to download the Fulcrum data for a release

Parameters

- **releases** – the IrusFulcrumRelease instance.
- **release** (IrusFulcrumRelease) –

upload_downloaded(*release*, ***kwargs*)

Upload the downloaded fulcrum data to the google cloud download bucket

Parameters
release (IrusFulcrumRelease) –

transform(*release*, ***kwargs*)

Task to transform the fulcrum data

Parameters
release (IrusFulcrumRelease) –

upload_transformed(*release*, ***kwargs*)

Upload the transformed fulcrum data to the google cloud download bucket

Parameters
release (IrusFulcrumRelease) –

bq_load(*release*, ***kwargs*)

Load the transfromed data into bigquery

Parameters
release (IrusFulcrumRelease) –

Return type
None

add_new_dataset_releases(*release*, ***kwargs*)

Adds release information to API.

Parameters
release (IrusFulcrumRelease) –

Return type
None

cleanup(*release*, ***kwargs*)

Delete all files and folders associated with this release.

Parameters
release (IrusFulcrumRelease) –

Return type
None

oaebu_workflows.workflows.irus_fulcrum_telescope.**download_fulcrum_month_data**(*download_month*, *requestor_id*, *num_retries=3*)

Download Fulcrum data for the release month

Parameters

- **download_month** (*pendulum.DateTime*) – The month to download usage data from
- **requestor_id** (*str*) – The requestor ID - used to access irus platform
- **num_retries** (*str*) – Number of attempts to make for the URL

Return type
Tuple[List[dict], List[dict]]

oaebu_workflows.workflows.irus_fulcrum_telescope.**transform_fulcrum_data**(*totals_data*, *country_data*, *publishers=None*)

Transforms Fulcrum downloaded "totals" and "country" data.

Parameters

- **totals_data** (*List[dict]*) – Fulcrum usage data aggregated over all countries
- **country_data** (*List[dict]*) – Fulcrum usage data split by country
- **publishers** (*List[str]*) – Fulcrum publishers to retain. If None, use all publishers

Return type
List[dict]

oaebu_workflows.workflows.irus_oapen_telescope

Module Contents

Classes

IrusOapenRelease	Create a IrusOapenRelease instance.
IrusOapenTelescope	The OAPEN irus uk telescope.

Functions

upload_source_code_to_bucket(source_url, project_id, ...)	Upload source code of cloud function to storage bucket
cloud_function_exists(service, full_name)	Check if cloud function with a given name already exists
create_cloud_function(service, location, full_name, ...)	Create cloud function.
call_cloud_function(function_uri, release_date, ...)	Iteratively call cloud function, until it has finished processing all publishers.

class oaebu_workflows.workflows.irus_oapen_telescope.**IrusOapenRelease**(*dag_id*, *run_id*, *data_interval_start*, *data_interval_end*, *partition_date*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Create a IrusOapenRelease instance.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID
- **partition_date** (*pendulum.DateTime*) – The date of the partition/release
- **data_interval_start** (*pendulum.DateTime*) –
- **data_interval_end** (*pendulum.DateTime*) –

class oaebu_workflows.workflows.irus_oapen_telescope.**IrusOapenTelescope**(*dag_id*, *cloud_workspace*, *publisher_name_v4*, *publisher_uuid_v5*, *data_partner='irus_oapen'*, *bq_dataset_description='IRUS dataset'*, *bq_table_description=None*, *api_dataset_id='oapen'*, *max_cloud_function_instances=0*, *observatory_api_conn_id=AirflowConns* *geoip_license_conn_id='geoip_license_k* *irus_oapen_api_conn_id='irus_api'*, *irus_oapen_login_conn_id='irus_login'*, *catchup=True*, *start_date=pendulum.datetime(2015, 6, 1)*, *schedule='0 0 4 * *'*, *max_active_runs=5*)

Bases: observatory.platform.workflows.workflow.Workflow

The OAPEN irus uk telescope. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param publisher_name_v4: The publisher's name for version 4 :param publisher_uuid_v5: The publisher's uuid for version 5 :param data_partner: The data partner :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param max_cloud_function_instances: :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param geoip_license_conn_id: The Airflow connection ID for the GEOIP license :param irus_oapen_api_conn_id: The Airflow connection ID for IRUS API - for counter 5 :param irus_oapen_login_conn_id: The Airflow connection ID for IRUS API (login) - for counter 4 :param catchup: Whether to catchup the DAG or not :param start_date: The start date of the DAG :param schedule: The schedule interval of the DAG :param max_active_runs: The maximum number of concurrent DAG instances

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **publisher_name_v4** (*str*) –
- **publisher_uuid_v5** (*str*) –
- **data_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **max_cloud_function_instances** (*int*) –
- **observatory_api_conn_id** (*str*) –
- **geoip_license_conn_id** (*str*) –
- **irus_oapen_api_conn_id** (*str*) –

- **irus_oapen_login_conn_id** (*str*) –
- **catchup** (*bool*) –
- **start_date** (*pendulum.DateTime*) –
- **schedule** (*str*) –
- **max_active_runs** (*int*) –

OAPEN_PROJECT_ID = 'oapen-usage-data-gdpr-proof'

OAPEN_BUCKET

FUNCTION_NAME = 'oapen-access-stats'

FUNCTION_REGION = 'europe-west1'

FUNCTION_SOURCE_URL =
'https://github.com/The-Academic-Observatory/oapen-irus-uk-cloud-function/releases/download/v1.1.8.

FUNCTION_MD5_HASH = '4bb8ab4ad8f31c93039f234b4d91cf3a'

FUNCTION_BLOB_NAME = 'cloud_function_source_code.zip'

FUNCTION_TIMEOUT = 1500

make_release(***kwargs*)

Create a list of IrusOapenRelease instances for a given month. Say the dag is scheduled to run on 2022-04-07 Interval_start will be 2022-03-01 Interval_end will be 2022-04-01 partition_date will be 2022-03-31

Parameters
kwargs – the context passed from the PythonOperator.

Return type
List[*IrusOapenRelease*]

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: list of IrusOapenRelease instances

transfer(*releases*, ***kwargs*)

Task to transfer the file for each release.

Parameters
releases (*List[IrusOapenRelease]*) – the list of IrusOapenRelease instances.

download_transform(*releases*, ***kwargs*)

Task to download the access stats to a local file for each release.

Parameters
releases (*List[IrusOapenRelease]*) –

create_cloud_function(*releases*, ***kwargs*)

Task to create the cloud function for each release.

Parameters
releases (*List[IrusOapenRelease]*) –

call_cloud_function(*releases*, ***kwargs*)

Task to call the cloud function for each release.

Parameters
releases (*List[IrusOapenRelease]*) –

upload_transformed(*releases*, ***kwargs*)

Uploads the transformed files to GCS for each release

Parameters
releases (*List[IrusOapenRelease]*) –

Return type
None

bq_load(*releases*, ***kwargs*)

Loads the sales and traffic data into BigQuery

Parameters
releases (*List[IrusOapenRelease]*) –

Return type
None

add_new_dataset_releases(*releases*, ***kwargs*)

Adds release information to API.

Parameters
releases (*List[IrusOapenRelease]*) –

Return type
None

cleanup(*releases*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
releases (*List[IrusOapenRelease]*) –

Return type
None

oaebu_workflows.workflows.irus_oapen_telescope.**upload_source_code_to_bucket**(*source_url*, *project_id*, *bucket_name*, *blob_name*, *cloud_function_path*)

Upload source code of cloud function to storage bucket

Parameters

- **source_url** (*str*) – The url to the zip file with source code
- **project_id** (*str*) – The project id with the bucket
- **bucket_name** (*str*) – The bucket name
- **blob_name** (*str*) – The blob name
- **cloud_function_path** (*str*) – The local path to the cloud function

Returns
Whether task was successful and whether file was uploaded

Return type
Tuple[bool, bool]

oaebu_workflows.workflows.irus_oapen_telescope.**cloud_function_exists**(*service*, *full_name*)

Check if cloud function with a given name already exists

Parameters

- **service** (*googleapiclient.discovery.Resource*) – Cloud function service
- **full_name** (*str*) – Name of the cloud function

Returns
URI if cloud function exists, else None

Return type
Optional[str]

oaebu_workflows.workflows.irus_oapen_telescope.**create_cloud_function**(*service*, *location*, *full_name*, *source_bucket*, *blob_name*, *max_active_runs*, *update*)

Create cloud function.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – Cloud function service
- **location** (*str*) – Location of the cloud function
- **full_name** (*str*) – Name of the cloud function
- **source_bucket** (*str*) – Name of bucket where the source code is stored
- **blob_name** (*str*) – Blob name of source code inside bucket
- **max_active_runs** (*int*) – The limit on the maximum number of function instances that may coexist at a given time
- **update** (*bool*) – Whether a new function is created or an existing one is updated

Returns
Status of the cloud function and error/success message

Return type
Tuple[bool, dict]

oaebu_workflows.workflows.irus_oapen_telescope.**call_cloud_function**(*function_uri*, *release_date*, *username*, *password*, *geoip_license_key*, *publisher_name_v4*, *publisher_uuid_v5*, *bucket_name*, *blob_name*)

Iteratively call cloud function, until it has finished processing all publishers. When a publisher name/uuid is given, there is only 1 publisher, if it is empty the cloud function will process all available publishers. In that case, when the data is downloaded from the new platform it can be done in 1 iteration, however for the old platform two files have to be downloaded separately for each publisher, this might take longer than the timeout time of the cloud function, so the process is split up in multiple calls.

Parameters

- **function_uri** (*str*) – URI of the cloud function
- **release_date** (*str*) – The release date in YYYY-MM

- **username** (*str*) – Oapen username (email or requestor_id)
- **password** (*str*) – Oapen password (password or api_key)
- **geoip_license_key** (*str*) – License key of geoip database
- **publisher_name_v4** (*str*) – URL encoded name of the publisher (used for counter version 4)
- **publisher_uuid_v5** (*str*) – UUID of the publisher (used for counter version 5)
- **bucket_name** (*str*) – Name of the bucket to store oapen access stats data
- **blob_name** (*str*) – Blob name to store oapen access stats data

Return type
None

oaebu_workflows.workflows.jstor_telescope

Module Contents

Classes

JstorRelease	Construct a JstorRelease.
JstorTelescope	The JSTOR telescope.

Functions

jstor_transform(download_country, ...)	Transform a Jstor release into json lines format and gzip the result._summary_
get_header_info(url)	Get header info from url and parse for filename and extension of file.
download_report(url, download_path)	Download report from url to a file.
get_release_date(report_path)	Get the release date from the "Reporting_Period" part of the header.
get_release_date_deprecated(report_path)	This function is deprecated, because the headers for the reports have changed since 2021-10-01.
create_gmail_service()	Build the gmail service.
get_label_id(service, label_name)	Get the id of a label based on the label name.
list_reports(service, publisher_id)	List the available releases by going through the messages of a gmail account and looking for a specific pattern.

class oaebu_workflows.workflows.jstor_telescope.**JstorRelease**(*dag_id*, *run_id*, *data_interval_start*, *data_interval_end*, *partition_date*, *reports_info*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Construct a JstorRelease.

Parameters

- **dag_id** (*str*) – The ID of the DAG

- **run_id** (*str*) – The Airflow run ID
- **data_interval_start** (*pendulum.DateTime*) – The beginning of the data interval
- **data_interval_end** (*pendulum.DateTime*) – The end of the data interval
- **partition_date** (*pendulum.DateTime*) – the partition date, corresponds to the last day of the month being processed.
- **reports_info** (*List[dict]*) – list with report_type (country or institution) and url of reports

class oaebu_workflows.workflows.jstor_telescope.**JstorTelescope**(*dag_id*, *cloud_workspace*, *publisher_id*, *country_partner='jstor_country'*, *institution_partner='jstor_institution'*, *bq_dataset_description='Data from JSTOR sources'*, *bq_country_table_description=None*, *bq_institution_table_description=None*, *api_dataset_id='jstor'*, *gmail_api_conn_id='gmail_api'*, *observatory_api_conn_id=AirflowConns.OBSERVA* *catchup=False*, *max_active_runs=1*, *schedule='0 0 4 * *'*, *start_date=pendulum.datetime(2016, 10, 1)*)

Bases: observatory.platform.workflows.workflow.Workflow

The JSTOR telescope.

Construct a JstorTelescope instance. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param publisher_id: The ID of the publisher for this DAG :param country_partner: The name of the country partner :param institution_partner: The name of the institution partner :param bq_dataset_description: Description for the BigQuery dataset :param bq_country_table_description: Description for the BigQuery JSTOR country table :param bq_institution_table_description: Description for the BigQuery JSTOR institution table :param api_dataset_id: The ID to store the dataset release in the API :param gmail_api_conn_id: Airflow connection ID for the Gmail API :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param catchup: Whether to catchup the DAG or not :param max_active_runs: The maximum number of DAG runs that can be run concurrently :param schedule: The schedule interval of the DAG :param start_date: The start date of the DAG

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **publisher_id** (*str*) –
- **country_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **institution_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_country_table_description** (*str*) –

- **bq_institution_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **gmail_api_conn_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **max_active_runs** (*int*) –
- **schedule** (*str*) –
- **start_date** (*pendulum.DateTime*) –

REPORTS_INFO = 'reports_info'

PROCESSED_LABEL_NAME = 'processed_report'

MAX_ATTEMPTS = 3

FIXED_WAIT = 20

MAX_WAIT_TIME

EXP_BASE = 3

MULTIPLIER = 10

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'.

Parameters
kwargs – the context passed from the PythonOperator.

Return type
List[*JstorRelease*]

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: A list of grid release instances

check_dependencies(***kwargs*)

Check dependencies of DAG. Add to parent method to additionally check for a publisher id

Returns
True if dependencies are valid.

Return type
bool

list_reports(***kwargs*)

Lists all Jstor releases for a given month and publishes their report_type, download_url and release_date's as an XCom.

Returns
Whether to continue the DAG

Return type
bool

download_reports(***kwargs*)

Download the JSTOR reports based on the list with available reports. The release date for each report is only known after downloading the report. Therefore they are first downloaded to a temporary location, afterwards the release info can be pushed as an xcom and the report is moved to the correct location.

Returns
Whether to continue the DAG (always True)

Return type
bool

upload_downloaded(*releases*, ***kwargs*)

Uploads the downloaded files to GCS for each release

Parameters
releases (*List[JstorRelease]*) – List of JstorRelease instances:

Return type
None

transform(*releases*, ***kwargs*)

Task to transform the Jstor releases for a given month.

Parameters
releases (*List[JstorRelease]*) –

upload_transformed(*releases*, ***kwargs*)

Uploads the transformed files to GCS for each release

Parameters
releases (*List[JstorRelease]*) –

Return type
None

bq_load(*releases*, ***kwargs*)

Loads the sales and traffic data into BigQuery

Parameters
releases (*List[JstorRelease]*) –

Return type
None

add_new_dataset_releases(*releases*, ***kwargs*)

Adds release information to API.

Parameters
releases (*List[JstorRelease]*) –

Return type
None

cleanup(*releases*, ***kwargs*)

Delete all files, folders and XComs associated with this release. Assign a label to the gmail messages that have been processed.

Parameters
releases (*List[JstorRelease]*) –

Return type
None

oaebu_workflows.workflows.jstor_telescope.**jstor_transform**(*download_country*, *download_institution*, *transfrom_country*, *transform_institution*, *partition_date*)

Transform a Jstor release into json lines format and gzip the result._summary_

Parameters

- **download_country** – The path to the country download report
- **download_institution** – The path to the institution download report
- **transfrom_country** – The path to write the transformed country file to
- **transform_institution** – The path to write the transformed institution file to
- **partition_date** – The partition/release date of this report

Return type
None

oaebu_workflows.workflows.jstor_telescope.**get_header_info**(*url*)

Get header info from url and parse for filename and extension of file.

Parameters
url (*str*) – Download url

Returns
Filename and file extension

Return type
List[str, str]

oaebu_workflows.workflows.jstor_telescope.**download_report**(*url*, *download_path*)

Download report from url to a file.

Parameters

- **url** (*str*) – Download url
- **download_path** (*str*) – Path to download data to

Return type
None

oaebu_workflows.workflows.jstor_telescope.**get_release_date**(*report_path*)

Get the release date from the "Reporting_Period" part of the header. Also checks if the reports contains data from exactly one month.

Parameters
report_path (*str*) – The path to the JSTOR report

Returns
The release date, defaults to end of the month

Return type
pendulum.DateTime

oaebu_workflows.workflows.jstor_telescope.**get_release_date_deprecated**(*report_path*)

This function is deprecated, because the headers for the reports have changed since 2021-10-01. It might still be used for reports that were created before this date and have not been processed yet. Get the release date from the "Usage Month" column in the first row of the report. Also checks if the reports contains data from the same month only.

Parameters
report_path (*str*) – The path to the JSTOR report

Returns
The start and end dates

Return type
pendulum.DateTime

oaebu_workflows.workflows.jstor_telescope.**create_gmail_service**()

Build the gmail service.

Returns
Gmail service instance

Return type
googleapiclient.discovery.Resource

oaebu_workflows.workflows.jstor_telescope.**get_label_id**(*service*, *label_name*)

Get the id of a label based on the label name.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – Gmail service
- **label_name** (*str*) – The name of the label

Returns
The label id

Return type
str

oaebu_workflows.workflows.jstor_telescope.**list_reports**(*service*, *publisher_id*)

List the available releases by going through the messages of a gmail account and looking for a specific pattern.

If a message has been processed previously it has a specific label, messages with this label will be skipped. The message should include a download url. The head of this download url contains the filename, from which the release date and publisher can be derived.

Parameters

- **service** (*googleapiclient.discovery.Resource*) – Gmail service
- **publisher_id** (*str*) – Id of the publisher

Returns
Dictionary with release dates as key and reports info as value, where reports info is a list of country

Return type
List[dict]

and/or institution reports.

oaebu_workflows.workflows.oapen_metadata_telescope

Module Contents

Classes

OapenMetadataRelease	Construct a OapenMetadataRelease instance
OapenMetadataTelescope	Oapen Metadata Telescope
OnixProduct	Represents a single ONIX product and its identifying reference for simplicity

Functions

download_metadata(uri, download_path)	Downloads the OAPEN metadata XML file
filter_through_schema(input, schema)	This function recursively traverses the input dictionary and compares it to the provided schema.
remove_invalid_products(input_xml, output_xml[, ...])	Attempts to validate the input xml as an ONIX file. Will remove any products that contain errors.
find_onix_product(all_lines, line_index)	Finds the range of lines encompassing a <Product> tag, given a line_number that is contained in the product

Attributes

DOWNLOAD_RETRY_CHAIN

oaebu_workflows.workflows.oapen_metadata_telescope.**DOWNLOAD_RETRY_CHAIN**

class oaebu_workflows.workflows.oapen_metadata_telescope.**OapenMetadataRelease**(*dag_id*, *run_id*, *snapshot_date*)

Bases: observatory.platform.workflows.workflow.SnapshotRelease

Construct a OapenMetadataRelease instance

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID
- **snapshot_date** (*pendulum.DateTime*) – The date of the snapshot_date/release

class oaebu_workflows.workflows.oapen_metadata_telescope.**OapenMetadataTelescope**(*dag_id*, *cloud_workspace*, *metadata_uri*, *metadata_partner='oapen_me*, *bq_dataset_description='OAP*, *Metadata converted to ONIX'*, *bq_table_description=None*, *api_dataset_id='oapen'*, *observatory_api_conn_id=Air*, *catchup=False*, *start_date=pendulum.datetime*, *5, 14)*, *schedule='@weekly'*)

Bases: observatory.platform.workflows.workflow.Workflow

Oapen Metadata Telescope

Construct a OapenMetadataTelescope instance. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param metadata_uri: The URI of the metadata XML file :param metadata_partner: The metadata partner name :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param catchup: Whether to catchup the DAG or not :param start_date: The start date of the DAG :param schedule: The schedule interval of the DAG

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **metadata_uri** (*str*) –
- **metadata_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **start_date** (*pendulum.DateTime*) –
- **schedule** (*str*) –

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'.

Parameters

kwargs – the context passed from the PythonOperator.

Return type

OapenMetadataRelease

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: The Oapen metadata release instance

download(*release*, ***kwargs*)

Task to download the OapenMetadataRelease release.

Parameters

- **kwargs** – the context passed from the PythonOperator.
- **release** (OapenMetadataRelease) – an OapenMetadataRelease instance.

Return type
None

upload_downloaded(*release*, ***kwargs*)

Task to upload the downloaded OAPEN metadata

Parameters
release (OapenMetadataRelease) –

Return type
None

transform(*release*, ***kwargs*)

Transform the oapen metadata XML file into a valid ONIX file This involves several steps 1) Parse the XML metadata to keep our desired fields 2) Remove products containing errors 3) Parse the validated ONIX file through the java parser to return .jsonl format 4) Add the contributor.personname field

Parameters
release (OapenMetadataRelease) –

Return type
None

upload_transformed(*release*, ***kwargs*)

Task to upload the transformed OAPEN metadata

Parameters
release (OapenMetadataRelease) –

Return type
None

bq_load(*release*, ***kwargs*)

Load the transformed ONIX file into bigquery

Parameters
release (OapenMetadataRelease) –

Return type
None

add_new_dataset_releases(*release*, ***kwargs*)

Adds release information to API.

Parameters
release (OapenMetadataRelease) –

Return type
None

cleanup(*release*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
release (OapenMetadataRelease) –

Return type
None

oaebu_workflows.workflows.oapen_metadata_telescope.**download_metadata**(*uri*, *download_path*)

Downloads the OAPEN metadata XML file OAPEN's downloader can give an incomplete file if the metadata is partially generated. In this scenario, we should wait until the metadata generator has finished. Otherwise, an attempt to parse the data will result in an XML ParseError. Another scenario is that OAPEN returns only a header in the XML. We want this to also raise an error. OAPEN metadata generation can take over an hour

Parameters

- **uri** (*str*) – the url to query for the metadata
- **download_path** (*str*) – filepath to store te downloaded file

Raises

- **ConnectionError** – raised if the response from the metadata server does not have code 200
- **AirflowException** – raised if the response does not contain any Product fields

Return type
None

oaebu_workflows.workflows.oapen_metadata_telescope.**filter_through_schema**(*input*, *schema*)

This function recursively traverses the input dictionary and compares it to the provided schema. It retains only the fields and values that exist in the schema structure, and discards any fields that do not match the schema.

 # Example usage with a dictionary and schema:

input_dict = {
"name": "John", "age": 30, "address": {

"street": "123 Main St", "city": "New York", "zip": "10001"

}

} schema = {

"name": null, "age": null, "address": {

"street": null, "city": null

}

} filtered_dict = filter_dict_by_schema(input_dict, schema) filtered_dict will be: {

"name": "John", "age": 30, "address": {

"street": "123 Main St", "city": "New York"

}

}

Parameters

- **input** (*dict*) – The dictionary to filter
- **schema** (*dict*) – The schema describing the desired structure of the dictionary

oaebu_workflows.workflows.oapen_metadata_telescope.**remove_invalid_products**(*input_xml*, *output_xml*, *invalid_products_file=None*)

Attempts to validate the input xml as an ONIX file. Will remove any products that contain errors.

Parameters

- **input_xml** (*str*) – The filepath of the xml file to validate
- **output_xml** (*str*) – The output filepath
- **invalid_products_file** (*str*) – The filepath to write the invalid products to. Ignored if unsupplied.

Return type
None

oaebu_workflows.workflows.oapen_metadata_telescope.**find_onix_product**(*all_lines*, *line_index*)

Finds the range of lines encompassing a <Product> tag, given a line_number that is contained in the product

Parameters

- **all_lines** (*list*) – All lines in the onix file
- **line_number** – The line number associated with the product
- **line_index** (*int*) –

Returns
A two-tuple of the start and end line numbers of the product

Raises
ValueError – Raised if the return would encompass a negative index, indicating the input line was not in a product

Return type
OnixProduct

class oaebu_workflows.workflows.oapen_metadata_telescope.**OnixProduct**

Represents a single ONIX product and its identifying reference for simplicity

product: dict

record_reference: str

oaebu_workflows.workflows.onix_telescope

Module Contents

Classes

OnixRelease	Construct an OnixRelease.
OnixTelescope	Construct an OnixTelescope instance.

class oaebu_workflows.workflows.onix_telescope.**OnixRelease**(*, *dag_id*, *run_id*, *snapshot_date*, *onix_file_name*)

Bases: observatory.platform.workflows.workflow.SnapshotRelease

Construct an OnixRelease.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID
- **snapshot_date** (*pendulum.DateTime*) – The date of the snapshot/release
- **onix_file_name** (*str*) – The ONIX file name.

class oaebu_workflows.workflows.onix_telescope.**OnixTelescope**(*, *dag_id*, *cloud_workspace*, *date_regex*, *sftp_root='/'*, *metadata_partner='onix'*, *bq_dataset_description='ONIX data provided by Org'*, *bq_table_description=None*, *api_dataset_id='onix'*, *observatory_api_conn_id=AirflowConns.OBSERVATO*, *sftp_service_conn_id='sftp_service'*, *catchup=False*, *schedule='@weekly'*, *start_date=pendulum.datetime(2021, 3, 28)*)

Bases: observatory.platform.workflows.workflow.Workflow

Construct an OnixTelescope instance. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param sftp_root: The working root of the SFTP server, passed to the SftoFolders class :param metadata_partner: The metadata partner name :param date_regex: Regular expression for extracting a date string from an ONIX file name :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param observatory_api_conn_id: Airflow connection ID for the overvatory API :param sftp_service_conn_id: Airflow connection ID for the SFTP service :param catchup: Whether to catchup the DAG or not :param schedule: The schedule interval of the DAG :param start_date: The start date of the DAG

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **date_regex** (*str*) –
- **sftp_root** (*str*) –
- **metadata_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **sftp_service_conn_id** (*str*) –

- **catchup** (*bool*) –
- **schedule** (*str*) –
- **start_date** (*pendulum.DateTime*) –

list_release_info(***kwargs*)

Lists all ONIX releases and publishes their file names as an XCom.

Parameters
kwargs – the context passed from the BranchPythonOperator.

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: the identifier of the task to execute next.

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'.

Returns
a list of Onix release instances.

Return type
List[*OnixRelease*]

move_files_to_in_progress(*releases*, ***kwargs*)

Move ONIX files to SFTP in-progress folder. :param releases: a list of Onix release instances

Parameters
releases (*List[OnixRelease]*) –

download(*releases*, ***kwargs*)

Task to download the ONIX releases.

Parameters
releases (*List[OnixRelease]*) –

upload_downloaded(*releases*, ***kwargs*)

Uploads the downloaded onix file to GCS

Parameters
releases (*List[OnixRelease]*) –

transform(*releases*, ***kwargs*)

Task to transform the ONIX releases.

Parameters
releases (*List[OnixRelease]*) –

upload_transformed(*releases*, ***kwargs*)

Uploads the transformed file to GCS

Parameters
releases (*List[OnixRelease]*) –

bq_load(*releases*, ***kwargs*)

Task to load each transformed release to BigQuery.

Parameters
releases (*List[OnixRelease]*) –

move_files_to_finished(*releases*, ***kwargs*)

Move ONIX files to SFTP finished folder.

Parameters
releases (*List[OnixRelease]*) –

add_new_dataset_releases(*releases*, ***kwargs*)

Adds release information to API.

Parameters
releases (*List[OnixRelease]*) –

Return type
None

cleanup(*releases*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
releases (*List[OnixRelease]*) –

Return type
None

oaebu_workflows.workflows.onix_work_aggregation

Module Contents

Classes

UnionFind	Union Find using weighted quick union and path compression.
BookWork	A book work is an abstract entity comprising the intellectual property embodied in a manifestation.
BookWorkFamily	A book work family aggregates different editions of a work together.
BookWorkAggregator	Aggregates ONIX records into "works" (BookWork object). If the WorkID exists in the ONIX record, it will use that.
BookWorkFamilyAggregator	Aggregates different editions of works into a family. The methodology will be similar to BookWorkAggregator.

Functions

get_pref_product_id(relprod)	Get the most preferred product ID. It will return the one with the lowest preference

class oaebu_workflows.workflows.onix_work_aggregation.**UnionFind**(*size*)

Union Find using weighted quick union and path compression. Instead of working on objects and requiring further implementation of comparators and size methods, this will operate on integers. Users should handle the mapping from objects to a distinct integer, e.g., by using the index of the objects if they are in a list. See: https://en.

wikipedia.org/wiki/Disjoint-set_data_structure and https://www.cs.princeton.edu/~rs/AlgsDS07/01UnionFind.pdf

Parameters
size (*int*) – Number of elements we are dealing with in total.

root(*node*)

Find the root (representative) of an object. Update root mappings along the way (path compression). :param node: Object to find the root for. :return: The object's root representative.

Parameters
node (*int*) –

Return type
int

find(*p*, *q*)

Check if two objects have the same root representative. :param p: First object. :param q: Second object. :return: Whether p, q have the same root representative.

Parameters

- **p** (*int*) –
- **q** (*int*) –

Return type
bool

unite(*p*, *q*)

Merge two objects into the same class, i.e., make the two objects have the same root representative. Use weighted union to merge smaller trees into the bigger trees. :param p First object. :param q Second object.

Parameters

- **p** (*int*) –
- **q** (*int*) –

get_partition()

Get the current class partition of the objects. :return: Partition of the objects as a list of lists (no guaranteed ordering).

Return type
List[List[int]]

class oaebu_workflows.workflows.onix_work_aggregation.**BookWork**(*, *work_id*, *work_id_type*, *products*)

A book work is an abstract entity comprising the intellectual property embodied in a manifestation. Works manifest themselves as products of different form, e.g., paperback, PDF.

Parameters

- **work_id** (*str*) – The Work ID.
- **work_id_type** (*str*) – Type scheme used in the Work ID.
- **products** (*List[Dict]*) – List of products that manifest the work.

add_product(*product*)

Add a product to the work. :param product: A product record.

Parameters
product (*dict*) –

class oaebu_workflows.workflows.onix_work_aggregation.**BookWorkFamily**(*, *works*, *work_family_id=None*, *work_family_id_type=None*)

A book work family aggregates different editions of a work together.

Parameters

- **works** (*List[BookWork]*) – List of works in the family.
- **work_family_id** – Work family ID.
- **work_family_id_type** (*Union[None, str]*) – Type of Work family ID.

oaebu_workflows.workflows.onix_work_aggregation.**get_pref_product_id**(*relprod*)

Get the most preferred product ID. It will return the one with the lowest preference number in the id_pref list, or an arbitrary identifier from the list if the listed preferences are not found. :param relprod: Related product record. :return: The Product ID type, and Product ID as a pair.

Parameters
relprod (*dict*) –

Return type
Tuple[str, str]

class oaebu_workflows.workflows.onix_work_aggregation.**BookWorkAggregator**(*records*)

Aggregates ONIX records into "works" (BookWork object). If the WorkID exists in the ONIX record, it will use that. The order of preference for the work identifier types is: ISBN-13 > DOI > Proprietary > everything else. If no identifier is specified in ONIX, i.e., no RelatedWorks info, one of the ISBN13 in the work will be used as a representative ID.

Parameters
records (*List[dict]*) – List of ONIX Product records.

filter_out_duplicate_records(*records*)

Filter out records with duplicate ISBNs. Logs the duplicates, and returns the filtered records.

Parameters
records (*dict*) – Product records.

Returns
Tuple of a list of filtered records, and a list of ISBNs which appear more than once in a record.

Return type
List[dict]

log_duplicate_isbns(*duplicates*)

Log the list of duplicate ISBNs encountered.

Parameters
duplicates (*List[str]*) – List of duplicate ISBNs.

get_pref_work_id(*identifiers*)

Tries to map the work identifier back to ISBN. :param identifiers: List of WorkIdentifiers. :return: Preferred work id type and the work id. None represents unknoown work_id type.

Parameters
identifiers (*List[Dict]*) –

Return type
Tuple[Union[None, str], Union[None, str]]

`aggregate()`

Run the aggregation process. Separate out the records into those containing RelatedWorks info, RelatedProducts info, and neithe. For a single publisher, this should only ever be 1 of the cases. Run different aggregation procedures for the 3 cases. If publishers are doing something funky, then we need to revisit this procedure. :return: List of BookWork objects representing the aggregated product records.

Return type
List[*BookWork*]

`is_relevant_work_relation(relwork)`

Check if the work relation code indicates a manifestation. :param relwork: Related work. :return: Whether the related work is a manifestation of the current work.

Parameters
relwork (*dict*) –

Return type
bool

`log_agg_relworks_errors(pisbn, wtype, wid)`

Log any errors from aggregating along RelatedWorks. :param pisbn: The product's ISBN. :param wtype: Type of WorkID. :param wid: WorkID of the related work. :return: True if we logged an error, False if it was OK.

Parameters

- **pisbn** (*str*) –
- **wtype** (*str*) –
- **wid** (*str*) –

`agg_relworks()`

Collect the entries with "Manifestation of" relation codes into a single work. The Work ID from that field will be used. This assumes that every product record has a "Manifestation of" field entry that either points to itself or something else. Revisit this if a publisher does it differently. :return: List of BookWork objects that categorise the product records.

Return type
List[*BookWork*]

`get_pid_idx(pid_type, pid)`

Get the product index (in self.records) using the Product ID information. :param pid_type: Product ID type. :param pid: Product ID. :return: Index to the product or None if record doesn't exist.

Parameters

- **pid_type** (*str*) –
- **pid** (*str*) –

Return type
Union[None, int]

`set_relevant_product_relation_codes()`

Returns
Set of relevant product codes indicating a manifestation of the current work.

Return type
Set[str]

`is_relevant_product_relation`(*relprod*)

Check if the related product has a relation indicating it's the same work as the current work. :param relprod: Related product. :return: Whether the product is a manifestation.

Parameters
relprod (*dict*) –

Return type
bool

`log_agg_related_product_errors`(*pisbn*, *relation*, *ptype*, *pid*)

Log the errors from aggregating along RelatedProducts.

Parameters

- **pisbn** (*str*) – The product's ISBN.
- **relation** (*str*) – The relation code.
- **ptype** (*str*) – Related product's identifer type.
- **pid** (*str*) – Related product's identifier.

`get_works_from_partition`(*partition*)

Convert the partition of equivalence classes of record indices into equivalence classes of BookWork objects. :param partition: Partition of product record indices as works equivalence classes. :return: Partition as BookWork objects.

Parameters
partition (*List[List[int]]*) –

Return type
List[*BookWork*]

`agg_relproducts`()

Aggregate the entries with targeted relation codes into a single work. Currently this is:

"Alternative format".

The Work ID will be an arbitrary ISBN13 representative from each work. :return: List of BookWork objects that categorise the product records.

Return type
List[*BookWork*]

`log_get_works_lookup_table_errors`(*manifestations*, *isbn*)

Log an error when an ISBN is assigned to multiple WorkIDs.

Parameters

- **manifestations** (*Set[str]*) – List of work IDs manifested by an ISBN.
- **isbn** (*str*) – ISBN that has multiple WorkID assignments.

`get_works_lookup_table`()

Aggregate the products into works, and output a list of dicts ready for jsonline conversion and BQ loading. Keys: ISBN, Work ID.

Returns
List of dicts.

Return type
List[dict]

class oaebu_workflows.workflows.onix_work_aggregation.**BookWorkFamilyAggregator**(*works*)

Aggregates different editions of works into a family. The methodology will be similar to BookWorkAggregator. This works with lists of BookWork objects rather than product records, so you need to have already aggregated products to works.

Parameters
works (*List[BookWork]*) – List of work objects.

set_relevant_product_codes()

Returns
Set of relevant product codes indicating different editions.

Return type
Set[str]

aggregate()

Run the aggregation process. Things that hint at edition information:

1. "Replaces", "Replaced by", "Is later edition of first edition" relation in RelatedProducts.
2. [Not implemented] "Derived from" and "Related work is derived from this" relation in RelatedWorks. This might need to be supplemented with other info.
3. [Not implemented] Title, Authors, EditionNumber.

Returns
List of book work families.

Return type
List[*BookWorkFamily*]

get_identifier_to_index_table(*identifier*)

Create a lookup table mapping identifiers to the index of the works list. :param identifier: Identifier name, e.g., ISBN13. :return: Lookup table.

Parameters
identifier (*str*) –

Return type
Dict

get_wid_idx(*pid_type*, *pid*, *isbn13_to_index*, *gtin13_to_index*, *proprietary_to_index*)

Get the index into the works list for the product id.

Parameters

- **pid_type** (*str*) – Product identifier type.
- **pid** (*str*) – Product ID.
- **isbn13_to_index** (*dict*) – ISBN lookup table.
- **gtin13_to_index** (*dict*) – GTIN13 lookup table.
- **proprietary_to_index** (*dict*) – Proprietary ID lookup table.

Returns
Index for the work in the works list, or None if the record doesn't exist.

Return type
Union[None, int]

`is_relevant_product_relation`(*relprod*)

Check whether a related product contains a code indicating different (equivalent content) editions. :param relprod: Related product. :return: Whether the related product is a different edition.

Parameters
relprod (*dict*) –

Return type
bool

`link_related_products`()

Partition the works into equivalence classes of work families based on product relation codes. :return: Partition of the works into work families (using work indices of the works list).

Return type
List[List[int]]

`agg_relproducts`()

Collect the entries with “Replaces”, “Replaced by”, “Is later edition of first edition” relation codes into a single family. The Work Family ID will be an arbitrary WorkID representative. :return: List of BookWork objects that categorise the product records.

Return type
List[*BookWorkFamily*]

`get_works_family_lookup_table`()

Aggregate the works into work families, and output a list of dicts ready for jsonline conversion and BQ loading. Keys: ISBN, Work family ID.

Returns
List of dicts.

Return type
Dict

oaebu_workflows.workflows.onix_workflow

Module Contents

Classes

OnixWorkflowRelease	Release information for OnixWorkflow.
OnixWorkflow	This workflow telescope:

Functions

dois_from_table(table_id[, doi_column_name, distinct])	Queries a metadata table to retrieve the unique DOIs. Provided the DOIs are not in a nested structure.
download_crossref_events(dois, start_date, end_date, ...)	Spawns multiple threads to download event data (DOI and publisher only) for each doi supplied.
download_crossref_event_url(url[, i])	Downloads all crossref events from a url, iterating through pages if there is more than one
download_crossref_page_events(url, headers)	Download crossref events from a single page
crossref_events_limiter()	"Task to throttle the calls to the crossref events API
transform_crossref_events(events[, max_threads])	Spawns workers to transforms crossref events
transform_event(event)	Transform the dictionary with event data by replacing '-' with '_' in key names, converting all int values to
create_latest_views_from_dataset(project_id, ...[, ...])	Creates views from all sharded tables from a dataset with a matching a date string.
get_onix_records(table_id)	Fetch the latest onix snapshot from BigQuery.
get_isbn_utils_sql_string()	Load the ISBN utils sql functions.

Attributes

CROSSREF_EVENT_URL_TEMPLATE

oaebu_workflows.workflows.onix_workflow.**CROSSREF_EVENT_URL_TEMPLATE** = **'https://api.eventdata.crossref.org/v1/events?mailto={mailto}&from-collected-date={start_date}&unt...'**

class oaebu_workflows.workflows.onix_workflow.**OnixWorkflowRelease**(*, *dag_id*, *run_id*, *snapshot_date*, *onix_snapshot_date*, *crossref_master_snapshot_date*)

Bases: `observatory.platform.workflows.workflow.SnapshotRelease`

Release information for OnixWorkflow.

Construct the OnixWorkflow Release :param dag_id: DAG ID. :param release_date: The date of the partition/release :param onix_snapshot_date: The ONIX snapshot/release date. :param crossref_master_snapshot_date: The release date/suffix of the crossref master table

Parameters

- **dag_id** (*str*) –
- **run_id** (*str*) –
- **snapshot_date** (*pendulum.DateTime*) –
- **onix_snapshot_date** (*pendulum.DateTime*) –
- **crossref_master_snapshot_date** (*pendulum.DateTime*) –

class oaebu_workflows.workflows.onix_workflow.**OnixWorkflow**(*dag_id*, *cloud_workspace*, *metadata_partner*, *bq_master_crossref_project_id='academic-observatory'*, *bq_master_crossref_dataset_id='crossref_metadata'*, *bq_oaebu_crossref_dataset_id='crossref'*, *bq_master_crossref_metadata_table_name='crossref_met*
bq_oaebu_crossref_metadata_table_name='crossref_meta
bq_crossref_events_table_name='crossref_events', *bq_country_project_id='oaebu-public-data'*, *bq_country_dataset_id='oaebu_reference'*, *bq_subject_project_id='oaebu-public-data'*, *bq_subject_dataset_id='oaebu_reference'*, *bq_book_table_name='book'*, *bq_book_product_table_name='book_product'*, *bq_oaebu_data_qa_dataset='oaebu_data_qa'*, *bq_oaebu_latest_data_qa_dataset='oaebu_data_qa_latse*
bq_onix_workflow_dataset='onix_workflow', *bq_oaebu_intermediate_dataset='oaebu_intermediate'*, *bq_oaebu_dataset='oaebu'*, *bq_oaebu_export_dataset='data_export'*, *bq_oaebu_latest_export_dataset='data_export_latest'*, *bq_worksid_table_name='onix_workid_isbn'*, *bq_worksid_error_table_name='onix_workid_isbn_errors*
bq_workfamilyid_table_name='onix_workfamilyid_isbn', *bq_dataset_description='ONIX workflow tables'*, *oaebu_intermediate_match_suffix='_matched'*, *data_partners=None*, *ga3_views_field='page_views'*, *schema_folder=default_schema_folder()*, *mailto='agent@observatory.academy'*, *crossref_start_date=pendulum.datetime(2018, 5, 14)*, *api_dataset_id='onix_workflow'*, *max_threads=2 * os.cpu_count() - 1*, *observatory_api_conn_id=AirflowConns.OBSERVATORY*
sensor_dag_ids=None, *catchup=False*, *start_date=pendulum.datetime(2022, 8, 1)*, *schedule='@weekly'*)

Bases: observatory.platform.workflows.workflow.Workflow

This workflow telescope: 1. [Not implemented] Creates an ISBN13-> internal identifier lookup table. 2. Creates an ISBN13 -> WorkID lookup table.

a. Aggregates Product records into Work clusters.

b. Writes the lookup table to BigQuery.

c. Writes an error table to BigQuery.

3. Create an ISBN13 -> Work Family ID lookup table. Clusters editions together.

a. Aggregate Works into Work Families.

b. Writes the lookup table to BigQuery.

4. Create OAEBU intermediate tables.

a. For each data partner, create new tables in oaebu_intermediate dataset where existing tables are augmented with work_id and work_family_id columns.

5. Create OAEBU QA tables for looking at metrics and problems arising from the data sets (and for eventual automatic reporting).

Initialises the workflow object.

Parameters

- **dag_id** (*str*) – DAG ID.
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) – The CloudWorkspace object for this DAG
- **bq_master_crossref_project_id** (*str*) – GCP project ID of crossref master data
- **bq_master_crossref_dataset_id** (*str*) – GCP dataset ID of crossref master data
- **bq_oaebu_crossref_dataset_id** (*str*) – GCP dataset ID of crossref OAeBU data
- **bq_master_crossref_metadata_table_name** (*str*) – The name of the master crossref metadata table
- **bq_oaebu_crossref_metadata_table_name** (*str*) – The name of the OAeBU crossref metadata table
- **bq_crossref_events_table_name** (*str*) – The name of the crossref events table
- **bq_country_project_id** (*str*) – GCP project ID of the country table
- **bq_country_dataset_id** (*str*) – GCP dataset containing the country table
- **bq_subject_project_id** (*str*) – GCP project ID of the subject tables
- **bq_subject_dataset_id** (*str*) – GCP dataset ID of the subject tables
- **bq_book_table_name** (*str*) – The name of the book table
- **bq_book_product_table_name** (*str*) – The name of the book product table
- **bq_oaebu_data_qa_dataset** (*str*) – OAEBU Data QA dataset.
- **bq_oaebu_latest_data_qa_dataset** (*str*) – OAEBU Data QA dataset with the latest data views
- **bq_onix_workflow_dataset** (*str*) – Onix workflow dataset.
- **bq_oaebu_intermediate_dataset** (*str*) – OAEBU intermediate dataset.
- **bq_oaebu_dataset** (*str*) – OAEBU dataset.
- **bq_oaebu_export_dataset** (*str*) – OAEBU data export dataset.
- **bq_oaebu_latest_export_dataset** (*str*) – OAEBU data export dataset with the latest data views
- **bq_worksid_table_name** (*str*) – table ID of the worksid table
- **bq_worksid_error_table_name** (*str*) – table ID of the worksid error table
- **bq_workfamilyid_table_name** (*str*) – table ID of the workfamilyid table
- **bq_dataset_description** (*str*) – Description to give to the workflow tables
- **oaebu_intermediate_match_suffix** (*str*) – Suffix to append to intermediate tables

- **data_partners** (*List[Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]]*) – OAEBU data sources.
- **ga3_views_field** – The name of the GA3 views field - should be either 'page_views' or 'unique_views'
- **schema_folder** (*str*) – the SQL schema path.
- **mailto** (*str*) – email address used to identify the user when sending requests to an API.
- **crossref_start_date** (*pendulum.DateTime*) – The starting date of crossref's API calls
- **api_dataset_id** (*str*) – The ID to store the dataset release in the API
- **max_threads** (*int*) – The maximum number of threads to use for parallel tasks.
- **observatory_api_conn_id** (*str*) – The connection ID for the observatory API
- **sensor_dag_ids** (*List[str]*) – Dag IDs for dependent tasks
- **catchup** (*Optional[bool]*) – Whether to catch up missed DAG runs.
- **start_date** (*Optional[pendulum.DateTime]*) – Start date of the DAG.
- **schedule** (*Optional[str]*) – Scheduled interval for running the DAG.
- **metadata_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –

make_release(***kwargs*)

Creates a release object. :param kwargs: From Airflow. Contains the execution_date. :return: an OnixWorkflowRelease object.

Return type
OnixWorkflowRelease

aggregate_works(*release*, ***kwargs*)

Fetches the ONIX product records from our ONIX database, aggregates them into works, workfamilies, and outputs it into jsonl files.

Parameters
release (OnixWorkflowRelease) –

upload_aggregation_tables(*release*, ***kwargs*)

Upload the aggregation tables and error tables to a GCP bucket in preparation for BQ loading.

Parameters
release (OnixWorkflowRelease) –

bq_load_aggregations(*release*, ***kwargs*)

Loads the 'WorkID lookup', 'WorkID lookup table errors' and 'WorkFamilyID lookup' tables into BigQuery.

Parameters
release (OnixWorkflowRelease) –

create_oaebu_crossref_metadata_table(*release*, ***kwargs*)

Creates the crossref metadata table by querying the AO master table and matching on this publisher's ISBNs

Parameters
release (OnixWorkflowRelease) – The onix workflow release object

create_oaebu_crossref_events_table(*release*, ***kwargs*)

Download, transform, upload and create a table for crossref events

Parameters
release (OnixWorkflowRelease) –

create_oaebu_book_table(*release*, ***kwargs*)

Create the oaebu book table using the crossref event and metadata tables

Parameters
release (OnixWorkflowRelease) –

create_oaebu_intermediate_table(*release*, ***, *orig_project_id*, *orig_dataset*, *orig_table*, *orig_isbn*, *sharded*, ***kwargs*)

Create an intermediate oaebu table. They are of the form datasource_matched<date> :param release: Onix workflow release information. :param orig_project_id: Project ID for the partner data. :param orig_dataset: Dataset ID for the partner data. :param orig_table: Table ID for the partner data. :param orig_isbn: Name of the ISBN field in the partner data table. :param sharded: Whether the data partner table is sharded

Parameters

- **release** (OnixWorkflowRelease) –
- **orig_project_id** (*str*) –
- **orig_dataset** (*str*) –
- **orig_table** (*str*) –
- **orig_isbn** (*str*) –
- **sharded** (*bool*) –

create_oaebu_book_product_table(*release*, ***kwargs*)

Create the Book Product Table

Parameters
release (OnixWorkflowRelease) –

export_oaebu_table(*release*, ***kwargs*)

Create an intermediate oaebu table. They are of the form datasource_matched<date>

Parameters
release (OnixWorkflowRelease) –

export_oaebu_qa_metrics(*release*, ***kwargs*)

Create the unmatched metrics table

Parameters
release (OnixWorkflowRelease) –

create_oaebu_export_tasks()

Create tasks for exporting final metrics from our OAEBU data. It will create output tables in the oaebu_elastic dataset.

create_oaebu_data_qa_tasks()

Create tasks for outputing QA metrics from our OAEBU data. It will create output tables in the oaebu_data_qa dataset.

create_oaebu_latest_views(*release*, ***kwargs*)

Create views of the latest data export tables in bigquery

Parameters
release (OnixWorkflowRelease) –

create_oaebu_data_qa_onix_aggregate(*release*, ***kwargs*)

Create a bq table of some aggregate metrics for the ONIX data set.

Parameters
release (OnixWorkflowRelease) –

create_oaebu_data_qa_isbn_onix(*release*, ***kwargs*)

Create a BQ table of invalid ISBNs for the ONIX feed that can be fed back to publishers. No attempt is made to normalise the string so we catch as many string issues as we can.

Parameters
release (OnixWorkflowRelease) –

oaebu_data_qa_validate_isbn(***, *orig_table_id*, *output_table_id*, *isbn*, *schema_file_path=None*)

Create a BQ table of invalid ISBNs for the ONIX feed that can be fed back to publishers. No attempt is made to normalise the string so we catch as many string issues as we can.

Parameters

- **orig_table_id** (*str*) – Fully qualified table ID of the source data
- **output_table_id** (*str*) – Fully qualified table ID for the output data.
- **isbn** (*str*) – Name of the isbn field in source table.
- **schema_file_path** (*str*) – The path of the schema file to use for the BigQuery upload

Returns
The status of the table creation

Return type
bool

create_oaebu_data_qa_isbn(*release*, *data_partner*, *table_name*, *isbn*, ***kwargs*)

Create a BQ table of invalid ISBNs for the Google Analytics feed. No attempt is made to normalise the string so we catch as many string issues as we can.

Parameters

- **release** (OnixWorkflowRelease) – workflow release object.
- **data_partner** (oaebu_workflows.oaebu_partners.OaebuPartner) – OaebuPartner,
- **table_name** (*str*) – The name of the table to create
- **isbn** (*str*) – Name of the isbn field in source table.

create_oaebu_data_qa_intermediate_unmatched_workid(*release*, *data_partner*, **args*, ***kwargs*)

Create quality assurance metrics for the OAEBU intermediate tables. :param data_partner: The OaebuPartner to use

Parameters

- **release** (OnixWorkflowRelease) –
- **data_partner** (oaebu_workflows.oaebu_partners.OaebuPartner) –

add_new_dataset_releases(*release*, ***kwargs*)

Adds release information to API.

Parameters
release (OnixWorkflowRelease) –

Return type
None

cleanup(*release*, ***kwargs*)

Cleanup temporary files.

Parameters
release (OnixWorkflowRelease) –

oaebu_workflows.workflows.onix_workflow.**dois_from_table**(*table_id*, *doi_column_name='DOI'*, *distinct=True*)

Queries a metadata table to retrieve the unique DOIs. Provided the DOIs are not in a nested structure.

Parameters

- **metadata_table_id** – The fully qualified ID of the metadata table on GCP
- **doi_field_name** – The name of the DOI column
- **distinct** (*str*) – Whether to retrieve only unique DOIs
- **table_id** (*str*) –
- **doi_column_name** (*str*) –

Returns
All DOIs present in the metadata table

Return type
List[str]

oaebu_workflows.workflows.onix_workflow.**download_crossref_events**(*dois*, *start_date*, *end_date*, *mailto*, *max_threads=1*)

Spawns multiple threads to download event data (DOI and publisher only) for each doi supplied. The url template was made with reference to the crossref event api: https://www.eventdata.crossref.org/guide/service/query-api/ Note that the max_threads will cap at 15 because the events API will return a 429 if more than 15 requests are made per second. Each API request happens to take roughly 1 second. Having more threadsthan necessary slows down the download process as the retry script will wait a minimum of two seconds between each attempt.

Parameters

- **dois** (*List[str]*) – The list of DOIs to download the events for
- **start_date** (*pendulum.DateTime*) – The start date for events we're interested in
- **end_date** (*pendulum.DateTime*) – The end date for events we're interested in
- **mailto** (*str*) – The email to use as a reference for who is requesting the data
- **max_threads** (*int*) – The maximum threads to spawn for the downloads.

Returns
All events for the input DOIs

Return type
List[dict]

oaebu_workflows.workflows.onix_workflow.**download_crossref_event_url**(*url*, *i=0*)

Downloads all crossref events from a url, iterating through pages if there is more than one

Parameters

- **url** (*str*) – The url send the request to
- **i** (*int*) – Worker number

Returns
The events from this URL

Return type
List[dict]

oaebu_workflows.workflows.onix_workflow.**download_crossref_page_events**(*url*, *headers*)

Download crossref events from a single page

Parameters

- **url** (*str*) – The url to send the request to
- **headers** (*dict*) – Headers to send with the request

Returns
The cursor, event counter, total number of events and the events for the URL

Return type
Tuple[str, int, int, List[dict]]

oaebu_workflows.workflows.onix_workflow.**crossref_events_limiter**()

"Task to throttle the calls to the crossref events API

oaebu_workflows.workflows.onix_workflow.**transform_crossref_events**(*events*, *max_threads=1*)

Spawns workers to transforms crossref events

Parameters

- **all_events** – A list of the events to transform
- **max_threads** (*int*) – The maximum number of threads to utilise for the transforming process
- **events** (*List[dict]*) –

Returns
transformed events, the order of the events in the input list is not preserved

Return type
List[dict]

oaebu_workflows.workflows.onix_workflow.**transform_event**(*event*)

Transform the dictionary with event data by replacing '-' with '_' in key names, converting all int values to string except for the 'total' field and parsing datetime columns for a valid datetime.

Parameters
event (*dict*) – The event dictionary

Returns
The transformed event dictionary

Return type
dict

oaebu_workflows.workflows.onix_workflow.**create_latest_views_from_dataset**(*project_id*, *from_dataset*, *to_dataset*, *date_match*, *data_location*, *description=None*)

Creates views from all sharded tables from a dataset with a matching a date string.

Parameters

- **project_id** (*str*) – The project id
- **from_dataset** (*str*) – The dataset containing the sharded tables
- **to_dataset** (*str*) – The dataset to contain the views
- **date_match** (*str*) – The date string to match. e.g. for a table named 'this_table20220101', this would be '20220101'
- **data_location** (*str*) – The regional location of the data in google cloud
- **description** (*str*) – The description for the views dataset

Return type
None

oaebu_workflows.workflows.onix_workflow.**get_onix_records**(*table_id*)

Fetch the latest onix snapshot from BigQuery. :param table_id: Fully qualified table ID. :return: List of onix product records.

Parameters
table_id (*str*) –

Return type
List[dict]

oaebu_workflows.workflows.onix_workflow.**get_isbn_utils_sql_string**()

Load the ISBN utils sql functions. :return BQ SQL string.

Return type
str

oaebu_workflows.workflows.thoth_telescope

Module Contents

Classes

ThothRelease	Construct a ThothRelease.
ThothTelescope	Construct an ThothOnixTelescope instance.

Functions

thoth_download_onix(publisher_id, download_path, ...)	Hits the Thoth API and requests the ONIX feed for a particular publisher.

Attributes

THOTH_URL
DEFAULT_HOST_NAME

oaebu_workflows.workflows.thoth_telescope.**THOTH_URL** = **'{host_name}/specifications/{format_specification}/publisher/{publisher_id}'**

oaebu_workflows.workflows.thoth_telescope.**DEFAULT_HOST_NAME** = **'https://export.thoth.pub'**

class oaebu_workflows.workflows.thoth_telescope.**ThothRelease**(*, *dag_id*, *run_id*, *snapshot_date*)

Bases: observatory.platform.workflows.workflow.SnapshotRelease

Construct a ThothRelease. :param dag_id: The ID of the DAG :param run_id: The Airflow run ID :param release_date: The date of the snapshot_date/release

Parameters

- **dag_id** (*str*) –
- **run_id** (*str*) –
- **snapshot_date** (*pendulum.DateTime*) –

class oaebu_workflows.workflows.thoth_telescope.**ThothTelescope**(*, *dag_id*, *cloud_workspace*, *publisher_id*, *format_specification*, *metadata_partner='thoth'*, *bq_dataset_description='Thoth ONIX Feed'*, *bq_table_description=None*, *api_dataset_id='onix'*, *host_name='https://export.thoth.pub'*, *observatory_api_conn_id=AirflowConns.OBSERVA*
catchup=False, *start_date=pendulum.datetime(2022, 12, 1)*, *schedule='@weekly'*)

Bases: observatory.platform.workflows.workflow.Workflow

Construct an ThothOnixTelescope instance. :param dag_id: The ID of the DAG :param cloud_workspace: The CloudWorkspace object for this DAG :param publisher_id: The Thoth ID for this publisher :param format_specification: The Thoth ONIX/metadata format specification. e.g. "onix_3.0::oapen" :param metadata_partner: The metadata partner name :param bq_dataset_description: Description for the BigQuery dataset :param bq_table_description: Description for the biguery table :param api_dataset_id: The ID to store the dataset release in the API :param host_name: The Thoth host name :param observatory_api_conn_id: Airflow

connection ID for the overvatory API :param catchup: Whether to catchup the DAG or not :param start_date: The start date of the DAG :param schedule: The schedule interval of the DAG

Parameters

- **dag_id** (*str*) –
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) –
- **publisher_id** (*str*) –
- **format_specification** (*str*) –
- **metadata_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) –
- **bq_dataset_description** (*str*) –
- **bq_table_description** (*str*) –
- **api_dataset_id** (*str*) –
- **host_name** (*str*) –
- **observatory_api_conn_id** (*str*) –
- **catchup** (*bool*) –
- **start_date** (*pendulum.DateTime*) –
- **schedule** (*str*) –

make_release(***kwargs*)

Creates a new Thoth release instance

Parameters
kwargs – the context passed from the PythonOperator.

Return type
ThothRelease

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: The Thoth release instance

download(*release*, ***kwargs*)

Task to download the ONIX release from Thoth.

Parameters
release (ThothRelease) – The Thoth release instance

Return type
None

upload_downloaded(*release*, ***kwargs*)

Upload the downloaded thoth onix XML to google cloud bucket

Parameters
release (ThothRelease) –

Return type
None

transform(*release*, ***kwargs*)

Task to transform the Thoth ONIX data

Parameters
release (ThothRelease) –

Return type
None

upload_transformed(*release*, ***kwargs*)

Upload the downloaded thoth onix .jsonl to google cloud bucket

Parameters
release (ThothRelease) –

Return type
None

bq_load(*release*, ***kwargs*)

Task to load the transformed ONIX jsonl file to BigQuery.

Parameters
release (ThothRelease) –

Return type
None

add_new_dataset_releases(*release*, ***kwargs*)

Adds release information to API.

Parameters
release (ThothRelease) –

Return type
None

cleanup(*release*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
release (ThothRelease) –

Return type
None

oaebu_workflows.workflows.thoth_telescope.**thoth_download_onix**(*publisher_id*, *download_path*, *format_spec*, *host_name=DEFAULT_HOST_NAME*, *num_retries=3*)

Hits the Thoth API and requests the ONIX feed for a particular publisher. Creates a file called onix.xml at the specified location

Parameters

- **publisher_id** (*str*) – The ID of the publisher. Can be found using Thoth GraphiQL API
- **download_path** (*str*) – The path to download ONIX the file to
- **format_spec** (*str*) – The ONIX format specification to use. Options can be found with the /formats endpoint of the API
- **host_name** (*str*) – The Thoth host URL

- **num_retries** (*int*) – The number of times to retry the download, given an unsuccessful return code

Return type
None

oaebu_workflows.workflows.ucl_discovery_telescope

Module Contents

Classes

UclDiscoveryRelease	Construct a UclDiscoveryRelease instance.
UclDiscoveryTelescope	The UCL Discovery telescope.

Functions

get_isbn_eprint_mappings(sheet_id, ...)	Get the eprint id to isbn mapping from the google sheet
download_discovery_stats(eprint_id, start_date, end_date)	Downloads the discovery stats for a given eprint ID within a specified date range.
transform_discovery_stats(country_record, ...)	Transforms the discovery stats for a single set of records

class oaebu_workflows.workflows.ucl_discovery_telescope.**UclDiscoveryRelease**(*dag_id*, *run_id*, *data_interval_start*, *data_interval_end*, *partition_date*)

Bases: observatory.platform.workflows.workflow.PartitionRelease

Construct a UclDiscoveryRelease instance.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **run_id** (*str*) – The Airflow run ID.
- **data_interval_start** (*pendulum.DateTime*) – The start of the data interval.
- **data_interval_end** (*pendulum.DateTime*) – The end of the data interval.
- **partition_date** (*pendulum.DateTime*) – The partition date for this release.

class oaebu_workflows.workflows.ucl_discovery_telescope.**UclDiscoveryTelescope**(*dag_id*, *cloud_workspace*, *sheet_id*, *data_partner='ucl_discovery'*, *bq_dataset_description='UCL Discovery dataset'*, *bq_table_description='UCL Discovery table'*, *api_dataset_id='ucl'*, *observatory_api_conn_id=Airflo* *oaebu_service_account_conn_id* *max_threads=os.cpu_count() * 2*, *schedule='0 0 4 * *'*, *start_date=pendulum.datetime(2 6, 1)*, *catchup=True*, *max_active_runs=10*)

Bases: observatory.platform.workflows.workflow.Workflow

The UCL Discovery telescope.

Construct a UclDiscoveryTelescope instance.

Parameters

- **dag_id** (*str*) – The ID of the DAG
- **cloud_workspace** (*observatory.platform.observatory_config.CloudWorkspace*) – The CloudWorkspace object for this DAG
- **sheet_id** (*str*) – The ID of the google sheet match eprint ID to ISBN13
- **data_partner** (*Union[str, oaebu_workflows.oaebu_partners.OaebuPartner]*) – The name of the data partner
- **bq_dataset_description** (*str*) – Description for the BigQuery dataset
- **bq_table_description** (*str*) – Description for the biguery table
- **api_dataset_id** (*str*) – The ID to store the dataset release in the API
- **observatory_api_conn_id** (*str*) – Airflow connection ID for the overvatory API
- **oaebu_service_account_conn_id** (*str*) – Airflow connection ID for the oaebu service account
- **max_threads** (*int*) – The maximum number threads to utilise for parallel processes
- **schedule** (*str*) – The schedule interval of the DAG
- **start_date** (*pendulum.DateTime*) – The start date of the DAG
- **catchup** (*bool*) – Whether to catchup the DAG or not
- **max_active_runs** (*int*) – The maximum number of concurrent DAG runs

make_release(***kwargs*)

Make release instances. The release is passed as an argument to the function (TelescopeFunction) that is called in 'task_callable'. There will only be 1 release, but it is passed on as a list so the SnapshotTelescope template methods can be used.

Parameters
kwargs – the context passed from the PythonOperator.

Return type
List[*UclDiscoveryRelease*]

See https://airflow.apache.org/docs/stable/macros-ref.html for the keyword arguments that can be passed :return: A list with one ucldiscovery release instance.

download(*release*, ***kwargs*)

Fownload the ucl discovery data for a given release. :param releases: The UCL discovery release.

Parameters
release (UclDiscoveryRelease) –

upload_downloaded(*release*, ***kwargs*)

Uploads the downloaded files to GCS

Parameters
release (UclDiscoveryRelease) –

transform(*release*, ***kwargs*)

Transform the ucl discovery data for a given release.

Parameters
release (UclDiscoveryRelease) –

upload_transformed(*release*, ***kwargs*)

Uploads the transformed file to GCS

Parameters
release (UclDiscoveryRelease) –

bq_load(*release*, ***kwargs*)

Loads the transformed data into BigQuery

Parameters
release (UclDiscoveryRelease) –

Return type
None

add_new_dataset_releases(*release*, ***kwargs*)

Adds release information to API.

Parameters
release (UclDiscoveryRelease) –

Return type
None

cleanup(*release*, ***kwargs*)

Delete all files, folders and XComs associated with this release.

Parameters
release (UclDiscoveryRelease) –

Return type
None

oaebu_workflows.workflows.ucl_discovery_telescope.**get_isbn_eprint_mappings**(*sheet_id*, *service_account_conn_id*, *cutoff_date*)

Get the eprint id to isbn mapping from the google sheet

Parameters

- **sheet_id** (*str*) – The ID of the google sheet.
- **credentials** – The credentials object to authenticate with.
- **cutoff_date** (*pendulum.DateTime*) – The cutoff date. If an item is published after this date, it will be skipped.
- **service_account_conn_id** (*str*) –

Return type
dict

oaebu_workflows.workflows.ucl_discovery_telescope.**download_discovery_stats**(*eprint_id*, *start_date*, *end_date*)

Downloads the discovery stats for a given eprint ID within a specified date range.

Parameters

- **eprint_id** (*str*) – The eprint ID of the item to get the stats for.
- **start_date** (*pendulum.DateTime*) – The start date of the date range.
- **end_date** (*pendulum.DateTime*) – The end date of the date range.

Returns
A tuple containing the country statistics and the total downloads statistics.

oaebu_workflows.workflows.ucl_discovery_telescope.**transform_discovery_stats**(*country_record*, *totals_record*, *isbn*, *title*)

Transforms the discovery stats for a single set of records

Parameters

- **country_record** (*dict*) – The country record
- **totals_record** (*dict*) – The totals record
- **isbn** (*str*) – The isbn that matches the eprint id
- **title** (*str*) –

Returns
The transformed stats

Return type
dict

Submodules

oaebu_workflows.airflow_pools

Module Contents

Classes

AirflowPool	Constructor an AirflowPool instance
CrossrefEventsPool	Constructor CrossrefEventsPool instance

class oaebu_workflows.airflow_pools.**AirflowPool**(*pool_name*, *pool_slots*, *pool_description*)

Constructor an AirflowPool instance

Parameters

- **pool_name** (*str*) – The name of this pool
- **pool_slots** (*int*) – The number of slots assigned to this pool
- **pool_description** (*str*) – A description of this pool

get_pool()

create_pool()

create_or_get_pool()

class oaebu_workflows.airflow_pools.**CrossrefEventsPool**(*pool_slots=15*)

Bases: *AirflowPool*

Constructor CrossrefEventsPool instance

Parameters

pool_slots (*int*) – The number of slots assigned to this pool

oaebu_workflows.api_type_ids

Module Contents

Classes

TableTypeId	TableTypeId type_id constants
DatasetTypeId	DatasetType type_id constants
WorkflowTypeId	WorkflowTypeId type_id constants

class oaebu_workflows.api_type_ids.**TableTypeId**

TableTypeId type_id constants

regular = 'regular'

sharded = 'sharded'

partitioned = 'partitioned'

class oaebu_workflows.api_type_ids.**DatasetTypeId**

DatasetType type_id constants

doab = 'doab'

oapen_metadata = 'oapen_metadata'

onix = 'onix'

google_analytics = 'google_analytics'

google_books_sales = 'google_books_sales'

google_books_traffic = 'google_books_traffic'

jstor_country = 'jstor_country'

jstor_institution = 'jstor_institution'

irus_oapen = 'irus_oapen'

ucl_discovery = 'ucl_discovery'

fulcrum = 'fulcrum'

onix_workflow = 'onix_workflow'

class oaebu_workflows.api_type_ids.**WorkflowTypeId**

WorkflowTypeId type_id constants

doab = 'doab'

oapen_metadata = 'oapen_metadata'

onix = 'onix'

thoth_onix = 'thoth_onix'

google_analytics = 'google_analytics'

google_books = 'google_books'

jstor = 'jstor'

irus_oapen = 'irus_oapen'

ucl_discovery = 'ucl_discovery'

fulcrum = 'fulcrum'

onix_workflow = 'onix_workflow'

oaebu_workflows.config

Module Contents

Functions

test_fixtures_folder(*subdirs)	Get the path to the Academic Observatory Workflows test data directory.
schema_folder()	Return the path to the database schema template folder.
sql_folder()	Return the path to the workflow SQL template folder.

oaebu_workflows.config.**test_fixtures_folder**(**subdirs*)

Get the path to the Academic Observatory Workflows test data directory.

Returns
the test data directory.

Return type
str

oaebu_workflows.config.**schema_folder**()

Return the path to the database schema template folder.

Returns
the path.

Return type
str

oaebu_workflows.config.**sql_folder**()

Return the path to the workflow SQL template folder.

Returns
the path.

Return type
str

oaebu_workflows.oaebu_partners

Module Contents

Classes

OaebuPartner	Class for storing information about data sources we are using to produce oaebu intermediate tables for.

Functions

partner_from_str(partner[, metadata_partner])	Get the partner from a string.

Attributes

OAEBU_METADATA_PARTNERS
OAEBU_DATA_PARTNERS

class oaebu_workflows.oaebu_partners.**OaebuPartner**

Class for storing information about data sources we are using to produce oaebu intermediate tables for.

Parameters

- **type_id** – The dataset type id. Should be the same as its dictionary key
- **bq_dataset_id** – The BigQuery dataset ID Bigquery Dataset ID.
- **bq_table_name** – The BigQuery table name Bigquery Table name
- **isbn_field_name** – Name of the field containing the ISBN.
- **title_field_name** – Name of the field containing the Title.
- **sharded** – whether the table is sharded or not.

type_id: str

bq_dataset_id: str

bq_table_name: str

isbn_field_name: str

title_field_name: str

sharded: bool

schema_path: str

__str__()

Return str(self).

oaebu_workflows.oaebu_partners.**OAEBU_METADATA_PARTNERS**

oaebu_workflows.oaebu_partners.**OAEBU_DATA_PARTNERS**

oaebu_workflows.oaebu_partners.**partner_from_str**(*partner*, *metadata_partner=False*)

Get the partner from a string.

Parameters

- **partner** (*Union[str,* OaebuPartner*]*) – The partner name.

- **metadata_partner** (*bool*) – If True, use the metadata partner dictionary; otherwise, use the data partners dictionary

Raises
Exception – Raised if the partner name is not found

Returns
The OaebuPartner

Return type
OaebuPartner

oaebu_workflows.onix

Module Contents

Classes

OnixParser	Class for storing infromation on the java ONIX parser

Functions

onix_collapse_subjects(onix)	The book product table creation requires the keywords (under Subjects.SubjectHeadingText) to occur only once
onix_create_personname_fields(onix)	Given an ONIX feed, attempts to populate the Contributors.PersonName and/or Contributors.PersonNameInverted
onix_parser_download([download_dir])	Downloads the ONIX parser from Github
onix_parser_execute(parser_path, input_dir, output_dir)	Executes the Java ONIX parser. Requires a .xml file in the input directory.

class oaebu_workflows.onix.**OnixParser**

Class for storing infromation on the java ONIX parser

Parameters

- **filename** – The name of the java ONIX parser file
- **url** – The url to use for downloading the parser
- **template** (*bash*) – The path to the bash template "onix_parser.sh.jinja2"

filename = 'coki-onix-parser-1.2-SNAPSHOT-shaded.jar'

url =
'https://github.com/The-Academic-Observatory/onix-parser/releases/download/v1.3.0/coki-onix-parser.

cmd = 'java -jar {parser_path} {input_dir} {output_dir}'

oaebu_workflows.onix.**onix_collapse_subjects**(*onix*)

The book product table creation requires the keywords (under Subjects.SubjectHeadingText) to occur only once Some ONIX feeds return all keywords as separate entires. This function finds and collapses each keyword into a semi-colon separated string. Other common separators will be replaced with semi-colons.

Parameters
onix (*List[dict]*) – The onix feed

Returns
The onix feed after collapsing the keywords of each row

Return type
List[dict]

oaebu_workflows.onix.**onix_create_personname_fields**(*onix*)

Given an ONIX feed, attempts to populate the Contributors.PersonName and/or Contributors.PersonNameInverted fields by concatenating the Contributors.NamesBeforeKey and Contributors.KeyNames fields where possible

Parameters
onix (*List[dict]*) – The input onix feed

Returns
The onix feed with the additional fields populated where possible

Return type
List[dict]

oaebu_workflows.onix.**onix_parser_download**(*download_dir=observatory_home('bin')*)

Downloads the ONIX parser from Github

Parameters
download_dir (*str*) – The directory to download the file to

Returns
(Whether the download operation was a success, The (expected) location of the downloaded file)

Return type
Tuple[bool, str]

oaebu_workflows.onix.**onix_parser_execute**(*parser_path*, *input_dir*, *output_dir*)

Executes the Java ONIX parser. Requires a .xml file in the input directory.

Parameters

- **parser_path** (*str*) – Filepath of the parser
- **input_dir** (*str*) – The input directory - first argument of the parser
- **output_dir** (*str*) – The output directory - second argument of the parser

Returns
Whether the task succeeded or not (return code 0 means success)

Return type
bool

O

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

O

P

R

S

T

U

V

W