

1. Getting Started with Zipline

- [1. Zipline Overview](#)
 - [Features](#)
 - [Role Based Access Control](#)
 - [Bulk Loading](#)
 - [Content Validations](#)
 - [Content Management](#)
 - [Downstream Deposits](#)
 - [API Integration](#)
- [Getting Started with Zipline](#)
- [Sign into Zipline](#)
 - [URL](#)
 - [Sign In](#)
 - [Creating a Zipline username and password to access the tool](#)
- [Log Out of Zipline](#)
 - [Log Out](#)
- [Zipline Landing Page](#)

1. Zipline Overview

Zipline is a content loading tool for loading journal articles and books content for clients on the newest version of the Silverchair Platform. The user interface allows for users to load content in zipped packages, preview their content, observe warnings and errors from their packages and schedule the content to be published. Upon publication, the content tool loading is configurable to deposit full text information or metadata to a wide range of downstream providers, including Crossref, PubMed Central, OVID, and others. The catalog section of the content loading tool allows users to filter and navigate to view details between different books or all articles present within a specific volume/issue of a journal.

Features

Below are some of the high level features available within the Zipline tool for all clients on the newest version of the Silverchair Platform:

Role Based Access Control

Clients can regulate and control access to certain aspects of the Zipline tool for qualified users. For example, a user with the Content Loading role can only view content they have individually loaded, while a user with the Content Production Manager role has additional permissions to view all content loaded to the tool and can make changes to content, like publishing newly loaded content.

Bulk Loading

Clients using the Zipline tool can choose to load content packages one at a time or queue up an entire set of content using an FTP (file transfer protocol).

Content Validations

Zipline offers a robust set of errors and warnings used to protect the integrity of content loaded in XML format. Each package loaded will display warnings and error messages (if applicable) on the overall structure, formatting, and use of features in the XML files.

Content Management

Users of Zipline have the ability to conduct a series of actions to the content, including but not limited to: previewing newly loaded content, publication, unpublishing articles, opening and closing issues, and scheduling a book for expiration/deletion.

Downstream Deposits

Once published, content loaded to Zipline can be sent to a variety of downstream deposit partners. The choice of submitting metadata or full text XML can be designated prior to enabling the downstream deposit service.

API Integration

Users of Zipline have the ability to upload, search for, and perform actions on content using the API interface if they prefer making API calls over using the UI. This can be particularly efficient when searching for specific information or performing a series of actions to the content.

Getting Started with Zipline

Sign into Zipline

Zipline can be accessed by any browser. For the most friendly experience, we recommend using either Chrome or Firefox.

URL

URLs are specific to the client using the tool. Please visit the URL provided by Silverchair and you can then sign in from any computer with proper credentials.

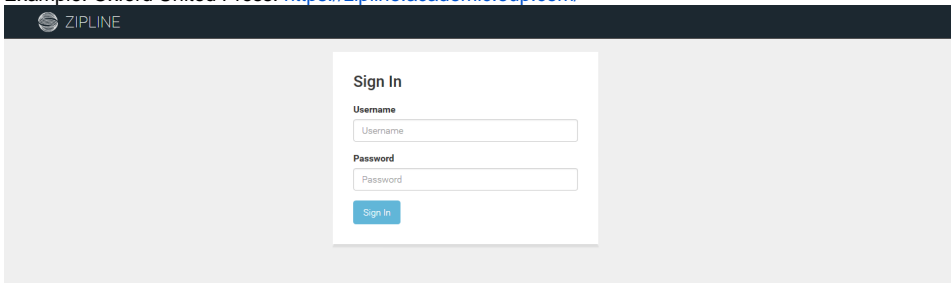
Sign In

1. Go to given URL
 - a. Below is a table describing the production URL for each client

i. Client	URL
OUP	https://zipline.academic.oup.com/
GSW	https://zipline.geoscienceworld.org/
DUP	https://zipline.read.dukeupress.edu/
IWA	https://zipline-iwa.silverchair.com/

2. Enter your given Username and Password Credentials

- Example: Oxford United Press: <https://zipline.academic.oup.com/>



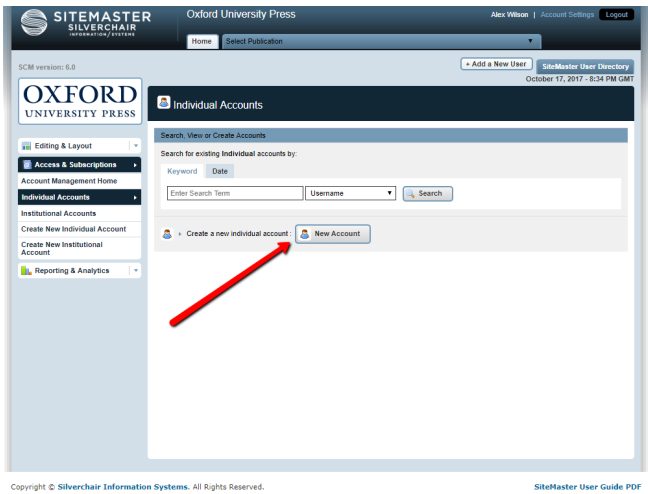
The screenshot shows the Zipline Sign In interface. At the top, there is a dark navigation bar with the Zipline logo and the word "ZIPLINE" in white. Below this, the main content area is light gray. In the center, there is a white rectangular box containing the "Sign In" form. The form has the title "Sign In" at the top. Below the title, there are two input fields: "Username" and "Password". Each field has a small placeholder text "Username" and "Password" respectively. At the bottom of the form is a blue button with the text "Sign In" in white.

Note: The 08 Mar 2018 release of Zipline adjusted the authentication page of Zipline to always redirect users to https if they attempt to access http first. This ensures the tool complies with more secure authentication standards used widely by websites and applications.

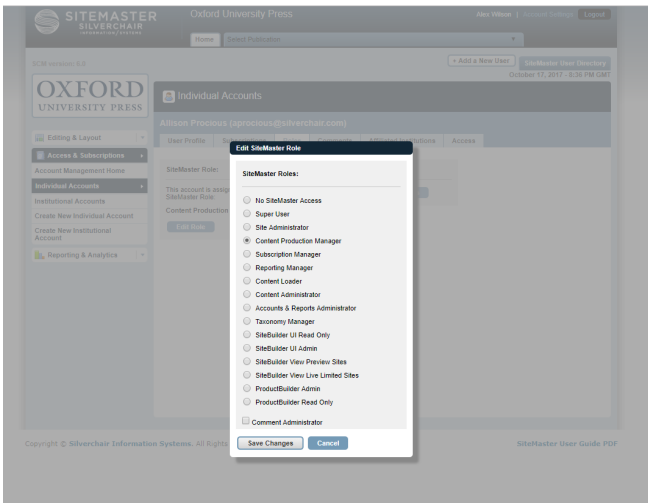
Creating a Zipline username and password to access the tool

If an individual needs to access the Zipline tool, anyone with the role of Super User can create new individual accounts for staff members and modify existing accounts via the Access & Subscriptions module in the SiteMaster application. Please keep in mind that you need to properly set up user roles in SiteMaster to create a new user account. For more information on SiteMaster, please reference that user guide for more details.

From the SiteMaster application, click 'Create a new individual account':



On the next page, enter the username, password (and confirm password), email address (and confirm address) of the new user. After creating the new user, navigate to the roles tab and click edit role:



Select the role that is right for the new user. A Zipline user needs to have one of the following roles assigned: Content Loader, Content Production Manager, Content Administrator, Taxonomy Manager, or Super User. For more information on these roles, please navigate to: [7. Access and User Roles](#). Once you have selected the role, press save changes. The new individual user will now be able to log into Zipline and access the tool, performing actions specific to their assigned role:

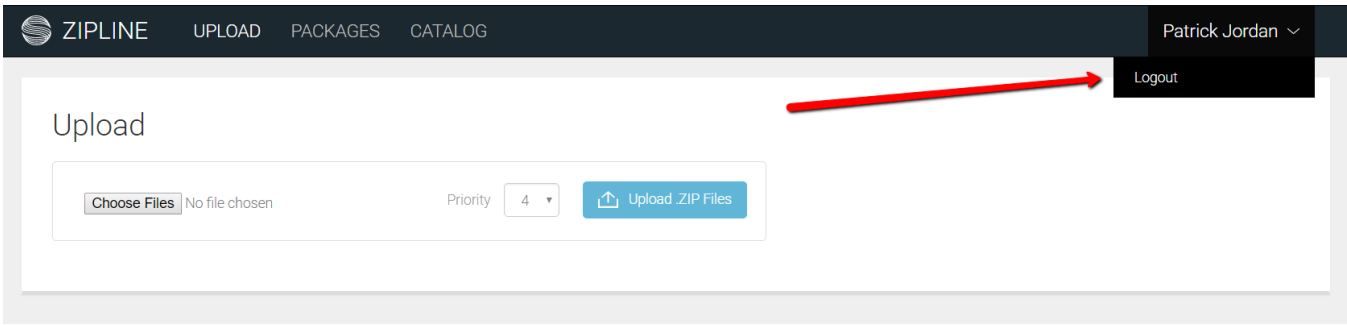
Log Out of Zipline

You can log out of Zipline from any interface once you have logged in.

Tip: Zipline will automatically log you out if you have been idle for 30 minutes.

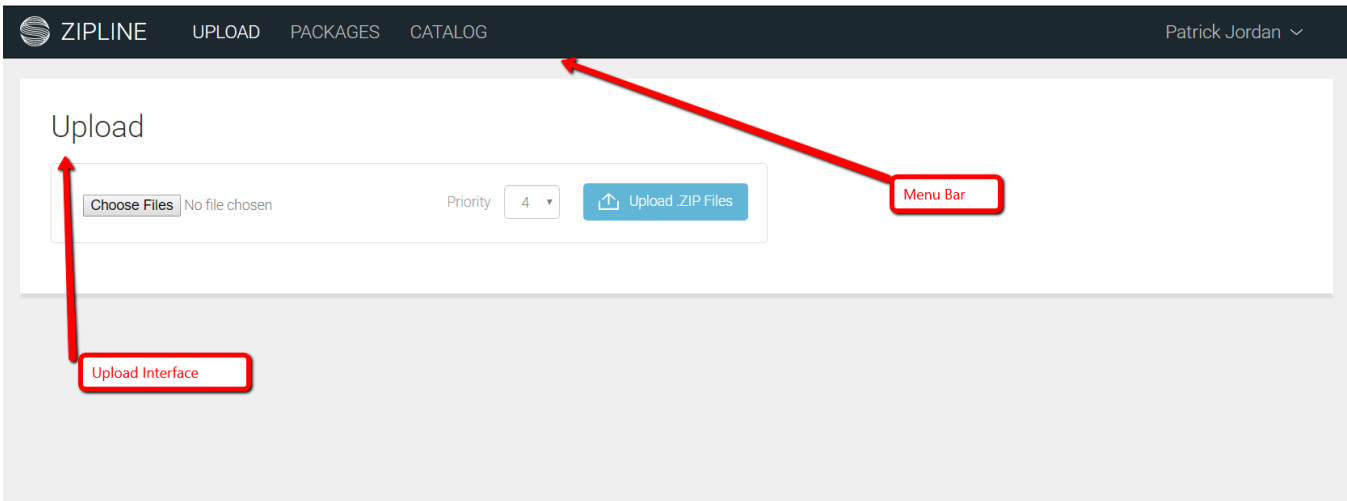
Log Out

1. In the top right corner, click your name
2. Click **Log Out**



Zipline Landing Page

Once logged into Zipline, you will have access to the landing page. The landing page is also the same as the Upload Page.



Menu Bar

The menu bar provides links for:

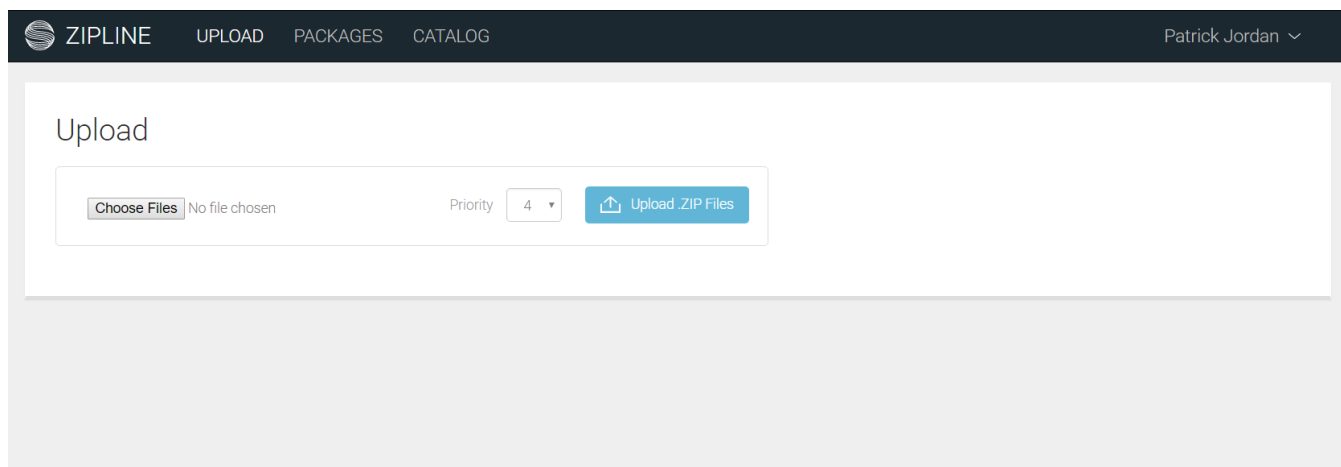
- Upload - allows users to upload content packages
- Packages - allows users to view status of loaded and in process packages
- Catalog - allows users to view and publish groups of content

2. Upload Content Packages

- [2. Uploading Content](#)
- [Zipline Upload Interface](#)
 - [Upload Interface Recent Changes](#)
- [FTP Interface](#)
- [Zipline Flowsteps Process](#)
- [Destination of Content Packages](#)
 - [Database Storage](#)
 - [Reference Enhancements](#)
 - [Scalability and Performance of Package Upload](#)
 - [Funding Information Associated with Journal Articles](#)
- [Package Requirements](#)
 - [ZIP Package File Size](#)
 - [ZIP Package File Name](#)
 - [ZIP Package Folder Structure](#)
 - [Note for Mac OS X Users](#)

2. Uploading Content

To upload zipped content packages to the Zipline tool, you can choose to select files via the UI (as seen below) or you can use the FTP to queue up an entire set of content to be loaded automatically. The file size limitations for uploading packages through the interface or the API is 1.5 GB. However, we offer the option to split a package into two or more separate ZIP files, where they can be loaded individually without issue.



Within the interface, users can select the relative priority of a select package against others currently in the queue, in order to more quickly process packages of higher priority. Zipline currently accepts zipped files with XML in SCJATS or SCBITS format. Journals content uses the SCJATS model, while books content will use the SCBITS format. To see more details on the Silverchair Platform Content Specifications for SCBITS and SCJATS, please reference the below specifications: documentation:

- [SCJATS](#)
- [SCBITS](#)

Zipline Upload Interface

Within the Upload Interface, these are three aspects of the user interface that you can interact with:

- **Choose Files:** This allows you to choose files to upload
- **Priority:** this dropdown menu allows you to choose the priority that the content should be upload on a scale of 0-9, with 9 as top priority in the system. Please reference the table on priority levels:

QueuePriorityLevelId	Name	Description
0	ScheduledBatch	Scheduled Batch
1	Batch	Batch (Default priority for submissions deposited to the FTP server.)
2	Low	Low
3	BelowNormal	Below Normal

4	Normal	Normal (Default priority for submissions made through the API.)
5	AboveNormal	Above Normal
6	High	High
7	Urgent	Urgent
8	Critical	Critical
9	Realtime	Realtime System Request

- Upload .ZIP Files: this button allows users to upload .ZIP packages that they have loaded

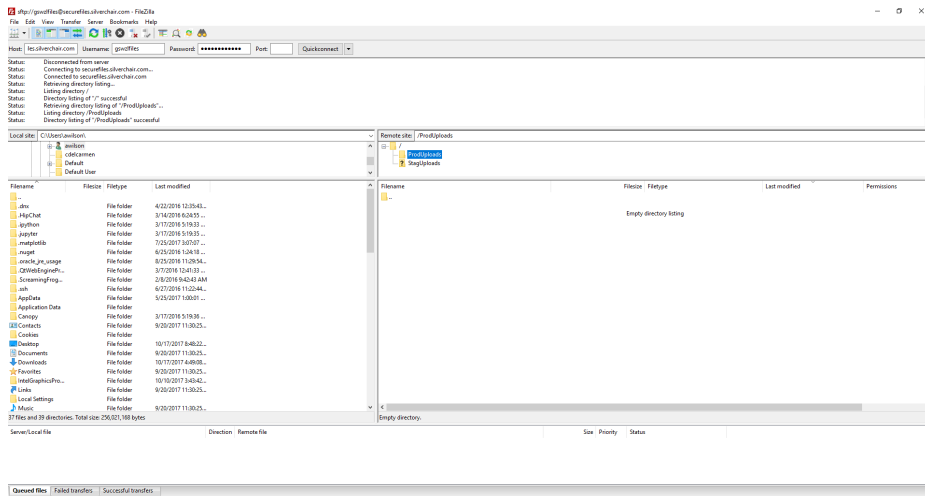
Upload Interface Recent Changes

- Queue Prioritization
 - The 19 Jul 2018 release to Zipline addresses improvements to Queue Prioritization of packages to ensure ZIP files processing in the system with the highest priority will be processed prior to other packages in the system. During times of peak usage, the queue prioritization feature allows for ZIP packages of the highest priority to be processed ahead of lower priority packages. During sparse usage (when less than 8 packages are loading at one time), the packages are processed sequentially (in the order they are loaded). This is due to the fact that there are 8 concurrently running instances of Zipline that pick up and process packages as they are loaded. Therefore, the prioritization feature is most useful during times of peak usage.
- Perform Upload Action:
 - The 19 Jul 2018 release to Zipline ensures that users are authenticated in Zipline before they attempt to upload a package. If their session has expired when they press the upload button, then they will be logged out of the tool immediately. If the user has a current session, then the package loading animation will occur normally.

FTP Interface

If users would prefer an alternative to uploading packages via the upload interface, they can use the FTP service to load a batch of ZIP files over the FTP. This may be beneficial in cases of bulk loading, where there are multiple packages that need to be loaded at one time. Another benefit to the FTP interface over using the Zipline UI or API is that there is no limit to the size of the ZIP files loaded with this method (whereas a limit of 1.5 GB is set on the API action). The WatchFTP service will draw from ZIP files loaded over the FTP on a schedule, and transfer them to the Zipline UI, where they will appear in process on the Packages page. One way that users can load packages to Zipline using the FTP is by connecting to FileZilla. The Bulk FTP loading tool is part of the suite of Zipline Content Tools.

- For all clients with the exception of Oxford University Press, the server is [sftp://securefiles.silverchair.com](ftp://securefiles.silverchair.com) and the credentials are only accessible via a support ticket to Client Services (or emailing support@silverchair.com)
- For clients using Zipline for Oxford University Press, the server is [sftp://oupsecurefiles.silverchair.com](ftp://oupsecurefiles.silverchair.com) and the credentials are only accessible via a support ticket to Silverchair Client Services

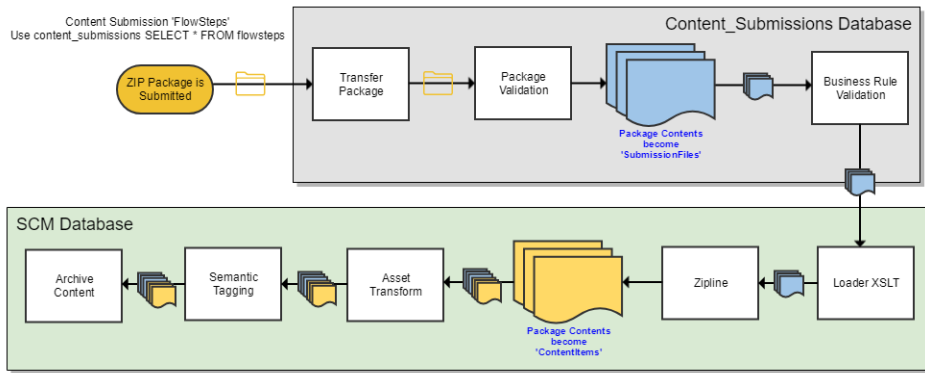


Accessing the FTP is only for temporary uploads to Zipline before the packages are processed. Due to storage restrictions, only 100GB can be loaded via the FTP at one time.

- The 24 May 2018 release solved an issue for some clients where the Watch FTP loader did not pick up new ZIP packages for processing. If you have any concerns about packages being picked up by the service, please do not hesitate to reach out to support@silverchair.com.

Zipline Flowsteps Process

Content submitted through Zipline follows a series of steps to full process each ZIP package. The steps range from validating the package structure, to importing sections of the XML into the database, storing images in Amazon Web Services, and archiving the content to be sent to downstream deposit services. For more information, please reference the following diagram:



Workflow Step	Description
Transfer Package	This workflow step uploads the content the client selected for loading and unzips it. The package is structured to have one XML folder and one Asset folder. The assets folder contains media files such as images and PDFs <ul style="list-style-type: none"> The 24 Jan 2019 newly began tracking the actions and time associated with this flowstep for throughput measurements
Package Validation	Does XML validation against an XML schema (such as SCJATS-journalpublishing.xsd or SCBITS-book.xsd) and makes sure the structure is valid.
Content Amalgamation	Optional, only specific to OxMeta files: An OxMetaProcessor that parses OxMeta file and update matching book/journal contents associated with OxMeta file.
Business Rule Validation	A service that tests against business rules, which could be Silverchair standards and/or rules that clients specify.
Loader XSLT	Transforms the XML contents so that it can be deserialized and stored in the database in the next flow step.
Zipline	Validates the data extracted from the transformed XML. If all tests pass, all the content will be written to the database. <ul style="list-style-type: none"> The 24 May 2018 release Refactored Zipline Section Loading for articles to ensure packages process more efficiently The 14 Jun 2018 release greatly reduced the number of database queries associated with the Zipline flowstep, enabling content XML to more efficiently be stored in the database.
Asset Transform	Transforms uncompressed type such as files ending with ".tif" into types that could be compressed such as files that end with ".png". Any XML references to the asset get changed so that they match the asset they reference. <ul style="list-style-type: none"> The 24 Jan 2019 release made improvements to this flowstep to use the UploadAsync method of AWS so that packages could load more efficiently The 21 Mar 2019 release made improvements to this flowstep to reduce the number of queries to the database when locating multimedia records and more efficiently copying asset files when loading them
Semantic Tagging	This step is optional. If the user chose this service, it makes a call to Semedica to import and autotag the content. If successfully imported and tagged, the semantic content will be staged for export to SCM6.
Archive Content	Stores all assets and XML to another section located in S3 ('ContentArchive'), so that it can be reused further, such as for downstream deposits. It also serves as a record so that it can be found by web crawlers such as Google Scholar.

- With the 14 Feb 2019 release to Zipline, we began tracking the system throughput average for Journals Content Submissions. The measurement calculation is Daily Minimum System Throughput Avg. (mbit/s) = Total daily size of compressed packages / Active processing seconds

Destination of Content Packages

Once uploaded to Zipline, packages travel to Amazon Web Services (AWS) S3 buckets (S3 stands for Simple Storage Service). The assets of a package, like PDF files, images, and figures are stored in the database with a specific path to their location in AWS, which is used by the site to serve the content to the end user to access. While the assets and the XML are divided out into specific locations within the S3 bucket structure based on the content type and other metadata, the original, unedited package is archived in AWS S3 and used for tracing a submission back to the original file. If you need assistance with locating a specific package loaded to Zipline, please contact Silverchair Client Services and we will locate the original ZIP file for you. Providing the submission key associated with the submission (see [4. Zipline Packages Page](#)) will greatly help with reducing the time needed to locate the original package.

- The 16 Nov 2017 release for Zipline further optimized the storage service tooling by removing redundant storage locations in AWS and the network file share. Once a package is loaded successfully to Zipline, AWS S3 serves as the system of record for a submission and the replicated ZIP file is removed from any internal network shares, allowing for more temporary file space for other services.
- The 25 Jan 2018 release to Zipline modified the Content Archive in AWS to store different parts of the book separately, which can then be used for metadata updates and depositing content to downstream providers. If a book is loaded as one package, the Content Archive separates that book into the book metadata in book-meta and the separate book part replaceable units (which include chapters, front matter sections, and book back sections). For more information on replaceable units, please reference this section: [4. Zipline Packages Page](#).
- The 15 Feb 2018 release to Zipline modified the Content Archive to store sourceXML reference for books, so that they can be used for metadata updates and display the raw XML for web crawlers
- The 08 Mar 2018 release to Zipline adjusted the Asset Transform flowstep of Zipline to store parts of books under the subfolder 'parts' rather than the subfolder 'chapters'. This was done because not all book parts are considered chapters. This change does not have any impact on the state of existing book parts in Zipline or the front end display of book parts.
- The 28 Mar 2018 release to Zipline resolved an issue where some PDFs loaded for book chapters were not available for download from the website. With this change, the PDFs were saved to the proper location in the content_public folder of AWS
- The 04 Oct 2018 release to Zipline ensured that the sourceXML destination in our database matches with the casing conventions of the XML in AWS, so that this content can be crawled by indexers
- The 25 Oct 2018 release to Zipline improved the content archive flowstep for books loading to more efficiently prepare books xml package for full text deposit services
- The 18 Dec 2018 release to Zipline ensures that the XML files in the content archive folder for books and book part wrappers files have asset references that match the assets in the asset folder. Extension of the files and the case should match. All references should be lowercase.
- The 24 Jan 2019 release to Zipline made an adjustment so that when a replaceable unit contains child sections below it, the content archive associates the child sections with the replaceable unit in AWS. This allow full text book deposits to function successfully
- The 21 Mar 2019 release to Zipline made an adjustment to the Content Archive service so that <collection-meta> is accurately stored in AWS when it's included in Book XML. This element <collection_meta> is within the original book XML. Previous it was not included because only <book-meta> and child nodes are archived in the sourceXml file, with a parent <book> wrapper element.
- The 11 Apr 2019 release to Zipline resolved an issue when loading a book-part-wrapper file in zipline 3. Previous, the asset transformer service did not save its assets to the correct S3 path. With the fix, the service now saves assets and ties them to the multimedia record in the database.

Database Storage

In addition to storage of the asset files in AWS, one of the Zipline workflow steps is to convert each element of the content's XML into specific database fields. This allows for a uniform presentation of content on the front end site, whether it belongs to a journal article, a book chapter page, the Issue Table of Contents, or the Book Table of Contents. The tagging structure in the XML will define how the content is imported into the database. For more information, please reference the [SCJATS](#) and the [SCBITS](#) narrative. Below are some details about the most recent releases that involved database changes to how content is stored:

- The 05 Oct 2017 release included logic that would translate how users tagged the license-type of their content. This change allows users to set and modify the access level of a book and associated book parts (whether they are free/not for sale/private) to determine how anonymous and subscribed users can view their content on the front end. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 16 Nov 2017 release modifies the Zipline tool to import abstracts with abstract-type="precis" in book-meta or book-part-meta
- The 04 Jan 2018 release adjusted the database storage logic so that books tagged with front-matter and book-back as free in the XML to be stored in the database as freely available to read. This ultimately allows users to access the content without authentication on the front end website. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 04 Jan 2018 release modified the Zipline tool to allow users to specify difference between author contributors and editor contributors for books. This will store them separately in the database, allowing them to be separated out on the front end book pages of the website. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details on the tagging structure.
- The 15 Feb 2018 release allows valid ISBN10 values with a check digit of X to continue Zipline validation and be imported into the database. ISBN10/ISBN13/EISBN values with digit other than numbers 0-9 or X still fails with error message code 50166. Please see [8. Zipline Error and Warning Codes](#) for more details.
- The 15 Feb 2018 release allows <publisher-name> values up to 255 characters to be imported successfully without truncation. If you find that your publisher-name exceed 255 characters, please contact your PDM (Product Delivery Manager) and we can develop a solution to meet your needs.
- The 15 Feb 2018 release adjusted Zipline to import supplementary material for books where the content-type is 'disclosure' or 'data-supplement'. It also modified the import process so that when a book is loaded to Zipline and there is supplementary-material with no content-type, the supplementary material defaults to data supplement. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 15 Feb 2018 release adjusted Zipline to import a book with a custom-meta-group designating it as an 'In Progress' book
- The 08 Mar 2018 release adjusted the database schema to prepare Zipline to import a book part with version information. New columns that will be used to store version information for a book include: Version Date, Version XML, Version Number, CurrentVersion ID, original publication date
- The 08 Mar 2018 release adjusted Zipline to import additional elements of edition information for a book. This will allow books tied by the same editionSet to be linked together, so that users can more easily navigate between them on the front end.
 - Order of the <collection-meta collection-type="edition-set"> among other collection-meta attributes does not matter, they will be imported and processed in any order they are supplied in the XML.
- The 08 Mar 2018 release adjusted Zipline to import book level publication dates in different locations, depending on if the value of publication-format was "ppub" or "epub". Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 28 Mar 2018 release adjusted Zipline to import affiliate logos to the database as book level supplementary material. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details

- The 28 Mar 2018 release adjusted Zipline to import the contents of book-app-group in book-back. Example XML snippet is below:

```

• <book-back>
  <book-app-group id="bid.300">
    <book-part-meta>
      <title-group>
        <title>TAXON Appendices</title>
      </title-group>
    </book-part-meta>
    <book-app id="bid.301">
      <book-part-meta>
        <title-group>
          <title>Appendix 1. TAXON nametypes.</title>
        </title-group>
      </book-part-meta>
      <body>
        <sec id="bid.302">
          <title>Scientific Name</title>
          <p>Every node in the database is required to have ...</p>
        </sec>
      </body>
    </book-app>
    <book-app id="bid.317">
      <book-part-meta>
        <title-group>
          <title>Appendix 2. Functional classes of TAXON scientific names.</title>
        </title-group>
      </book-part-meta>
      <body></body>
    </book-app>
    <book-app id="bid.331">
      <book-part-meta>
        <title-group>
          <title>Appendix 3. Other TAXON data types.</title>
        </title-group>
      </book-part-meta>
      <body></body>
    </book-app>
  </book-app-group>
</book-back>

```

- The 03 May 2018 release adjusted Zipline to import supplementary material at the chapter level for books content. This change allows content to be stored in the database and have the asset stored within AWS

- Example tagging for a book part:

```

385 <supplementary-material id="B001-sup100" content-type="data-supplement" xlink:href="data_supplement100.pdf">
386 <object-id pub-id-type="doi">10.9999/SSC_2012092100_v10713_043_sup1</object-id>
387 <label>Supplementary Material</label>
388 <caption>
389 <p>Detailed diagram of the topic under discussion.</p>
390 </caption>
391 </supplementary-material>
...

```

- The 24 May 2018 release allows users to set issue publication date and issue cover date when loading an issue meta file for the first time, or update the issue publication date/issue cover date for an existing issue. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
 - The order of importing issue publication date from the issue meta is <pub-date pub-type="collection"> if present, and then <pub-date pub-type="ppub"> if not present
 - The order of importing issue cover date from the issue meta is <pub-date pub-type="cover"> if present, and then <pub-date pub-type="collection">, and then <pub-date pub-type="ppub"> is neither are present
- The 24 May 2018 release allows users to specify license-type="cc-by-igo" in their XML to allow the access type to be stored in the database for the article. This license type will make the content available on the site without authentication. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 24 May 2018 release adjusted content import for users to include Sketchfab 3D models in their XML. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 14 Jun 2018 release allows users to specify license-type="publisher-standard" in their XML to include with paywalled articles. This enables submission and discovery of the license information on Crossref. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 09 Aug 2018 release allows users to optionally include abstract-type="video", abstract-type="audio", content-type="featured-video", and content-type="featured-audio" in their XML. This allows featured video/audio content and video/audio abstracts to be stored in the database for journals articles. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 04 Sep 2018 release allows users to set issue title when loading an issue meta file for the first time, or update the issue title for an existing issue. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details.
- The 04 Sep 2018 release allows users to import a publisher location value for a book, known in SCBITS as <publisher-loc>. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details

```
• <publisher>
  <publisher-name>Kluwer Academic Publishers</publisher-name>
  <publisher-loc>Norwell, MA</publisher-loc>
</publisher>
```

- The 04 Oct 2018 release allows users to load a book or a book part wrapper file with a fig nod that has a graphic with license-type="NoElectronicRight" to designate a restricted image. Users will not include this image in the package load to Zipline, but instead this indication will be stored in the database so that sites configured to display it will provide a message to users stating that they must purchase the print book to view the image. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 25 Oct 2018 release allows users to import chapter level keywords for books content where the kwd-group-type is a value other than "author". A user loading a package can specify a value other than "author" when loading author assigned keywords for a book chapter. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 25 Oct 2018 release allows users to specify <issue-sponsor> in either article meta or issue meta to specify who might have sponsored the issue. Please see the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details
- The 15 Nov 2018 release to Zipline resolved an issue where fig-group/caption/p was not being imported/saved to the database for books content. Going forward all books loaded with this detail will have the information stored and appropriately displayed where configured for a client site.
- The 24 Jan 2019 release to Zipline imported collection-type="collection" into the database as a new category, so that we can import information supplied in SCBITS

```
• <collection-meta collection-type="collection"> <title-group> <title>OXMEDO</title> </title-group>
</collection-meta>
```

- In the 14 Feb 2019 release to Zipline, the following elements of Zipline storage were adjusted:

- Introduced two new author types in the system, chair and translator. Users can specify these values in the XML to designate authors as a chair or a translator. Clients who have specific author types that they would like configured in addition to the standard types accepted by Crossref should contact Silverchair Client Services and their PDM directly.
- A regression was resolved that enabled books to be imported with book series with either of the following XML formats:

```
• <collection-meta>
  <title-group>
    <title>Book Series</title>
  </title-group>
</collection-meta>

<collection-meta collection-type="series">
  <title-group>
    <title>Book Series</title>
  </title-group>
</collection-meta>
```

- In the 21 Mar 2019 release to Zipline, the following elements of Zipline storage were adjusted:

- Importing of <author-comment> in the <contrib> element for books content into the database. The following content-types are supported:
 - <author-comment content-type="disclosure">
 - <author-comment content-type="url">
 - <author-comment content-type="occupation">
 - <author-comment content-type="other">
 - <author-comment content-type="deceased">
- Importing values for book-part-type in front-matter-part and book-part in book back, in order to distinguish between front matter and back matter pieces in a search filter on the front end
 - Values do not need to be pre-configured, they can be stored in the database based on the value in the XML
- Importing preview PDFs for monographs (books authored wholly by the same contributors) to be indexed by Google Scholar
 - The client may choose to generating the Preview PDFs in the format that Google Scholar will accept
 - Then these can be loaded into the database and displayed on the front end site with configuration from the team supporting the books site
- Importing book type for book loaded to Zipline, to differentiate between edited volumes and monographs
 - Chapters from edited volumes - in which each chapter is authored by folks not the editor(s) - function very much like journal articles. Each chapter is able to "stand alone" for indexing and be disassociated from the overall volume without compromising the user experience in search results.
 - Paywall monographs, however, due to their make up as mature works & separate user-experience, do not fit within the full-text indexing model for journals. For the monographs - which again would focus on the "preview pdfs" for restricted-access - the setups would be slightly different from other models:
- Import self-uri in book-part-meta
 - This enables users to load end notes PDFs in book-back to Zipline so that they will display in the proper location in the back matter of a book
- Import keywords in book-meta
 - This enables users to specify book level keywords in the book meta so that they can be associated with the book

- In the 11 Apr 2019 release to Zipline, the following elements of Zipline storage were adjusted:

- An issue was resolved that prevented contributors with <collab> tagging from being imported to Zipline
- Resolved issue that prevented Arabic characters in a book title from being displayed. The cause was due to the use of varchar in the storage of the book's title. Changing to nvarchar resolved the issue.

- Support 'indexed' attribute on front-matter-part and book-back. This enables content to be optionally excluded from search
- Support for importing book indexes into the database
- In the 02 May 2019 release to Zipline, the following elements of Zipline storage were adjusted:
 - Silverchair now distinguishes between similar sections that exist in both front-matter and book-back. This enables a clear way to determine which parent node the data came from originally, and allows for sorting of these section the front end. The parts that are similar are listed below:
 - Acknowledgement
 - //book/front-matter/ack
 - //book/book-back/ack
 - AuthorExtra
 - //book/front-matter/bio
 - //book/book-back/bio
 - Dedication
 - //book/front-matter/dedication
 - //book/book-back/dedication
 - FootNoteGroup
 - //book/front-matter/fn-group
 - //book/book-back/fn-group
 - Books with XML /book/book-back/dedication is saved to the database and all assets are saved to S3
 - Enabled all child elements under <named-book-part-body> to be parsed correctly into the database
 - Resolved issue that incorrectly assigned heading levels to chapters and parts
 - Imported inline-graphic when placed inside of related-object to the database
 - Imported related-object where an external link is referenced in the book or book part. This creates a link in the database to be displayed on the front end website if configured appropriately
 - Imported related-object where multimedia is referenced as the destination
 - Import authors with a contrib-type value that is configured for the the client's Eligible Author Types
 - Import <table-wrap-group> for books
 - Import <related-object> for books and book parts with book and section as a destination
 - Import <related-object> with page number specified
- In the 06 Jun 2019 release to Zipline:
 - Adjust import logic to create multimedia records in the database when there is <inline-graphic> inside of <title> for books
- In the 27 Jun 2019 release to Zipline:
 - When an article abstract is loaded with the specific-use="layperson" attribute, the abstract's section is imported with a classType of "layperson". This enables the front end display of sites with the appropriate configurations to differentiate between authored abstracts and layperson abstracts
 - Updated the related links process of link articles to other articles upon Zipline loading. Logic is as follows:
 - When an article is loaded with a related-article link
 - And the link contains an (elocation-id OR the link contains ext-link-type="doi" and xlink:href="[DOI]") and related-article-type
 - And the destination did NOT exist in our database when the source article was loaded
 - And the destination exists in the database now
 - Then a sql job is run on a schedule to create a link to the now existing destination article
 - And the related-article link contains ALL of the following: (journal-id-type="issn" OR journal-id-type="eissn") and journal-id and vol and page and related-article-type
 - And the destination did NOT exist in our database when the source article was loaded
 - And the destination exists in the database now
 - Then a sql job is run on a schedule to create a link to the now existing destination article
 - Otherwise a link is not created
- In the 18 Jul 2019 release to Zipline:
 - Imported the designation equal-contrib="yes" for contributors that contributed equally to a given piece of content. This enables the front end site to designate which authors contributed equally and display a footnote including more information.
 - Imported the fn-type="COI statement" associated with authors where a conflict of interest is specified. This enables the front end site to designate the author foot note surrounding conflict of interest information

Reference Enhancements

As part of package loading, Zipline queries external third party integrations such as Pubmed and Crossref to locate external links that aid in content discoverability. This enables references at the end of an article or book section to contain links to each reference to browse on Pubmed and Crossref. The Reference Enhancements stored procedure uses API calls to send queries to content in external databases and retrieve data sufficient to create the external link to the site. If a result is returned from the API call, then Silverchair stores the identifying information in the database along with the reference. Sometimes the information in the reference is not detailed enough to locate a valid reference and perform the enhancement. In this case, the reference is still stored in the database, but the link to the external site is not returned. This can be resolved by including additional data in the reference and reloading the package. Silverchair recommends following the latest Minor Version of the SCBITS and SCJATS Narrative Specifications for more details

The 14 Jun 2018 release resolved an issue with Pubmed reference enhancements where the API request was longer than acceptable to return a result. To resolve, the POST command was used, enabling Pubmed reference enhancements to be added to article references.

The 09 Aug 2018 release to Zipline began including Pubmed API Keys, in accordance with a Pubmed directive to utilize API keys when querying the Pubmed API (see more details here: <https://ncbiinsights.ncbi.nlm.nih.gov/2018/04/19/testing-periods-for-new-api-keys/>). On September 1st, 2018, Pubmed will be limiting queries via a single API key to no more than 10 requests per second. Each client on the Silverchair Platform has a separate API key to ensure there are no gaps in performance as a result of this change.

The 04 Sep 2018 release to Zipline set up additional Pubmed API Keys for new clients MIT Press and ASM International, in accordance with the previous note.

Scalability and Performance of Package Upload

The Silverchair Platform includes continuous enhancements to improve user and publisher experiences on Zipline. In addition to new features, this includes ongoing bug fixes, stability modifications, and optimization. The 05 Oct 2017 release included refactoring and optimizations for loading assets to AWS to eliminate timeouts tied to package uploading. The 07 Dec 2017 release to Zipline resolved an issue on production where some additional timeouts caused fatal errors to occur on the package upload Content Archive and Asset Transform flowstep. The resolution also enabled the package details page of uploaded package to load more quickly for users, providing a better overall experience. The 25 Oct 2018 release to Zipline adjusted a function when loading books content that greatly improved overall performance during periods of large backfile loading.

If you have issues with loading a valid package to Zipline, please consult with Silverchair Client Services (providing the submission key associated with the package) and we can resolve any issue for you.

Funding Information Associated with Journal Articles

- This information is available in the [SCJATs narrative](#)
- To include information about funding of the research, please use the <funding-group> tag in the XML. Zipline can import the following elements of the content and deposit it to Crossref:
 - funder name
 - funder identifier
 - Award identifiers
- Inside the <funding-source> element, use the <named-content content-type="funder-name"> and <named-content content-type="funder-identifier"> tags to designate the funder name and identifiers, respectively.
- If an embargo period is stored in the database for a funder, articles associated with an award from that funder will automatically become available outside the paywall (free) after that embargo period. Embargo periods for funding agencies are maintained in the database and are not included in the XML.
 - This feature was released with the 24 May 2018 production post
 - To view the complete embargo periods for funding agencies, please navigate to this link: <https://www.chorusaccess.org/resources/us-agency-public-access-plans-details/>
 - To view the list of funders on the Open Funder Registry, please navigate to: <https://support.crossref.org/hc/en-us/articles/214360886-The-Open-Funder-Registry>
 - If a funder supplied in the XML does not match the list of funders on the Open Funder Registry, then a warning message will be displayed and the funding information will not be imported.
 - Please reference MessageCode 50190 on this page [8. Zipline Error and Warning Codes](#) for more information
- The initial access type for an article is set during import. A license URL for an article can also be sent to CrossRef. See section [Permissions](#) for information about setting access type and including licensing information.

Package Requirements

SCJATS and SCBITS (journals and books) content must be supplied in ZIP packages conforming to the standards herein in order to be imported through Zipline.

ZIP packages must follow the directory structure

Files in the ZIP package must be one of the accepted file types (more details in [3. ZIP Package Contents](#))

References to asset files located in the assets folder must be contained within the XML for the files to be imported and associated with the content

While package may contain XML files for both books and journals in the same ZIP file, Silverchair recommends dividing content of a single content type into separate packages

ZIP Package File Size

The maximum file size for a .ZIP package is capped at 1.5 G. Uploading files greater than 1 GB carries a higher risk of the transfer failing due to time or file corruption. If a package is greater than 1 G, consider breaking the package up into two or more smaller chunks of content.

ZIP Package File Name

The ZIP file has no requirements except that the extension must be **.ZIP**. It is recommend the file name be something that is meaningful to the publisher and representative of the content (e.g. "JABCv99i9_20150908_1"). For example, a package that represents a publish ahead of print ('PAP') article could have the name 'pap' in the name, while a package representing a book could have the name 'book' or the ISBN in the name of the package.

ZIP Package Folder Structure

ZIP files must have the following folder structure:

The XML and Assets folders must be at the top level of the ZIP file. For example, the XML folder may not be located in the ZIP file at the path /A/XML. A package will fail if it does not contain an XML folder, or if there is no XML file in the XML folder.

Folder names within the ZIP package must be all one word with no spaces.

There must be one XML folder in the ZIP package, and there must be at least one XML file within it.

If no asset files exist, the assets folder is not needed.

Folder Name	Required or Optional	Folder Contents
XML	Required	XML files that are primary content: article XML (where the root element is <article>) issue-meta XML (where the root element is <issue-meta>) book XML (where the root element is <book>) book part XML (where the root element is <book-part-wrapper>)
Assets	Optional	All files referenced by the XML that are imported with the content, including data supplements that are XML files

Note for Mac OS X Users

The Mac OS X file system may create hidden folders in the ZIP file that prevent our system from finding content. To ensure ZIP files created on in OS X import, create ZIP files from inside the base folder by multi-selecting the XML folder and Assets folder directly. Errors regarding .DS_Store files can be ignored.

3. ZIP Package Contents

- [3. Package Contents Overview](#)
- [XML Contents of ZIP Packages](#)
 - [Articles](#)
 - [Journal Articles](#)
 - [Journal issue-level files](#)
 - [Issue cover image caption](#)
 - [Issue cover image](#)
 - [Proceeding articles](#)
 - [Replacements](#)
 - [Books](#)
- [Assets](#)
 - [File Naming Convention](#)
 - [Accepted File Types](#)
 - [Content PDFs](#)
 - [Book Level Asset Files](#)
 - [Link to PDF of content](#)
 - [Image specifications](#)
 - [File format](#)
 - [Color space](#)
 - [File size and quality](#)

3. Package Contents Overview

The Zipline tool is configured to validate a series of packaging rules and checks against the XML. These checks are necessary to enforce the package requirements on the Silverchair Platform. Each requirement is set with data integrity in mind and ease to make updates to content as it is loaded to Zipline. For both articles and books loaded to Zipline, a general convention of each .ZIP file is to include separate folders for assets and XML. Zipline accepts a wide range of file types for assets and connects each file to specific content items via references to the asset in the XML file.

XML Contents of ZIP Packages

Articles

Journal Articles

For Journal content, each ZIP file must contain content from the same issue, or only articles of the same life-cycle type. A ZIP file containing Publish Ahead of Print (PAP) articles cannot also contain articles for an issue, nor can it contain any Just Accepted Manuscripts (JAM). A ZIP file containing an issue must contain files only for that single issue.

When a new issue contains the final print version of articles formerly published as Publish Ahead of Print or Just Accepted Manuscript, these articles should be included with the issue. Both the PAP or JAM and the final version of the article must have the same DOI so that the system can match the final version with the PAP or JAM version. The system will replace the PAP or JAM with the final version upon publication of the issue. For more information on replacements, please reference the visual diagram on [19. Zipline Visual Diagrams](#).

Journal issue-level files

Issue-level files such as the cover, front matter, back matter must be present in the **Assets** folder to be imported and attached to the issue. The files must also be identified in the Issue Manifest XML file. If issue-level files are already associated to an issue, and another import contains an issue-manifest file and issue-level files, the new import will replace the files from the previous import. However, new imports will not delete previously imported issue-level files.

Issue cover image caption

Captions to cover images can be included in a text file. The text file must be a simple text file, not rich text.

Issue cover image

Assets for issue cover images must be supplied in one of the following file formats: .bmp, .gif, .jpeg, .tif, .tiff, or .png. If the file format for an issue cover image does not match one of the accepted formats (like a .pdf), then the image will not display.

Proceeding articles

Proceeding article ZIP files may contain:

- A single proceeding
- Multiple proceedings belonging to the same volume
- Multiple proceedings belonging to different volumes within the same conference
- Multiple proceedings belonging to different volumes for more than one conference

The system will check to see if the conference hierarchy exists for each proceeding and create the conference hierarchy if it does not.

Replacements

Replacing an article requires that Zipline receives the entire article, including assets (images, tables, PDF) and XML; in other words, all files associated with the article must be included in the ZIP package.

Books

A ZIP package must contain either a single book file, or multiple book part files, but cannot mix both ("book part" here is used loosely to mean any chapter- and part-like content, including a Part, Chapter, Section, Foreword, etc.). In other words, including two or more xml files with the root element of <book> will cause the package to fail. If a Zipline user needs to upload two separate books, they can split them into different .ZIP packages. If the package contains book part files (xml with a root element of <book-part-wrapper>, all files must be from the same book. If the book part files reference different books, all files will fail Zipline validation. However, book parts for a single book need not be supplied in a single ZIP package; they may be split across multiple ZIP packages for a given book.

To view details on what elements on a book file are imported or additional details around the contents of a book's XML, please reference the SCBITS Narrative. You can view the latest minor version on [Zipline User Guide](#).

Assets

All assets files must be referenced from content XML. Asset files are files other than the primary content—article, book, or book part—XML. These files include figure images, videos, audio, data supplements, issue cover images, book cover images, and other issue- and book-level ancillary files. Assets may include XML files; for example, a data supplement that is an XML file should be included in the Assets folder, not the XML folder.

File Naming Convention

Depending on the content type, asset file names must be unique at a certain level, specified in the following table. Even if two asset files have different extensions, their filenames should still be unique (i.e. do not name one file 123.tiff and another 123.png):

Content	File name uniqueness
journal articles in an issue	All asset files must be uniquely named within an issue.
journal articles not in issue	All asset files must be uniquely named across all PAP (OLF) and JAM articles for that journal.
proceeding articles	All asset files must be uniquely named within a conference session.
book parts	All asset files must be uniquely named within a book.

Silverchair suggests that publishers follow a file naming convention that helps identify content of the file. Silverchair does not require any particular naming convention. However, following a naming convention such as the example below can help aid in troubleshooting content loading and can help ensure unique file names across the content.

Content	Suggested File Name Convention Example
Articles	journal_vol_issue_firstpage.xml
Figures	journal_vol_issue_firstpage_f001.jpg
Data Supplements	journal_vol_issue_firstpage_ds001.pdf
Article PDFs	journal_vol_issue_firstpage.pdf
Disclosures	journal_vol_issue_firstpage_d.pdf

Accepted File Types

The following table lists the currently accepted file types for Assets files. They are cross referenced to the asset material that they can be used for. If you have files of a type that is not in this list, please contact Silverchair to have them added. Silverchair will not permit the following executable files in packages: BAT, CMD, EXE, BIN, DLL.

	Asset Material

File Type	File Formats Accepted	Content PDF (self-uri)	Data Supplements (supplementary-material)	Disclosures (supplementary-material)	Figure, Table, Graphics content	Issue-Level Files	Book-Level Files
PDF	<ul style="list-style-type: none"> pdf 	✓	✓	✓	-	✓	✓
Images	<ul style="list-style-type: none"> jpg jpeg* png gif** tif tiff bmp 	-	✓	✓	✓	✓	✓
Video	<ul style="list-style-type: none"> mpg mpeg mov avi wmv 	-	✓	✓	✓	✓	✓
Audio	<ul style="list-style-type: none"> mp3 mp4 wav au 	-	✓	✓	✓	✓	✓
Other file types accepted as of current date:	.db,.doc,.docx,.fla,.flv,.htm,.html,.m4a,.pdf.png,.ppt,.pptx,.sgm,.sgml,.swf,.txt,.wma,.xls,.xlsx,.xml,.zip	-	✓	✓	✓	✓	✓
Other file types accepted 11/7/2019)	ab1, adz, ai, aif, aiff, aln, applescript, asf, atom, bed, bib, bw, c, cab, cdf, cdr, cel, cellml, cif, class, cp, cpp, css, csv, cvx, cys, dmg, dnd, do, doc, docm, docx, dotm, dotx, dta, dtl, dvi, edf, edrw, emf, ens, eps, exe, f, f4v, fas, fasta, fidbundle, flv, fna, fsa, fst, gff, gif, gz, hmm, hmr, hqx, htm, html, htslp, jar, java, jnb, js, kml, kmz, llb, lsm, lut, m, m1v, m4a, m4v, mat, matlab, mdb, mdl, meg, mmd, mml, mp2, mpga, msf, msi, msw, miz, mw, nb, net, nex, nxs, ode, ogv, onetoc, os, pbm, pdb, pgm, phy, pir, pl, potm, potx, ppam, ppm, pps, ppsm, ppsx, ppt, pptm, pptx, ps, psd, pse, py, qt, r, ra, rar, rdata, rm, rtf, rv, sbml, sgml, shar, shtml, sit, skip, sldm, sldx, spt, svg, swf, tar, tex, thmx, tsv, txt, vmd, webarchive, webm, wig, wma, wpd, wrl, xlam, xls, xlsx, xism, xlsx, xlt, xltm, xltx, xml, xsl, xyz, zip, 7z	-	✓	✓	-	✓	✓

* The “jpeg” extension is required for JPGs displaying on Amazon Kindle.

** GIF files may not have a transparent background.

Content PDFs

If your content has a corresponding PDF, use print quality PDFs.

Book Level Asset Files

Books may have ancillary material at the book level such as a book cover, front matter, etc. that publishers want available on the site in non-html formats. To add book-level assets to the book, use the <supplementary-material> tag in the <book-meta>. PDFs must be used as the file extension for book supplementary material, with the exception of book cover images. For the cover image, other file extensions may be used such as PNG.

Link to PDF of content

To import a PDF and link it to an article or book/book part, use the <self-uri> element. Only file extension of PDF should be used with the <self-uri> element.

Image specifications

File format

Silverchair’s preferred web format is PNG-24, but all other formats listed above are acceptable.

- TIFF (and TIF) files submitted are converted to PNG-24 at the same size; all other formats retain their originally submitted types and specifications.

- JPG images are transformed to JPEG format
- EPS files cannot be loaded through Zipline, and must be converted to a web-based format or TIFF prior to loading.

Aside from the conversion of TIFF to PNG and JPG to JPEG, images submitted remain in their original state and become the default view/download for the large versions on the website. The package XML is updated within AWS with references to the updated extensions as well (which is used in downstream deposits).

Color space

Submitted web formats retain their original color space (Index, RGB, Grey) when loaded through Zipline.

TIFFs are converted to RGB as their format is converted to PNG. For this reason, TIFFs in the bitmap color mode will result in an undesirably large file size. It is highly recommended that the color space for both bitmap and CMYK TIFFs be converted to RGB and the files resized before loading through Zipline.

File size and quality

Image files loaded through Zipline should be of ideal web viewing size, since originals remain unchanged and are the default for the "View Large" window. There is no automated resizing of original image files. Files at print resolution will retain their resolution at full size and may hinder page-load time and usability. Likewise, image files that are too small for adequate web viewing will be further compromised at their smaller sizes.

The loading process automatically creates medium- and small-sized versions of the original images for display in appropriate windows on the page. The loading process also saves the filenames for all content types with lowercase formatting to the database and transforms the XML reference in Amazon Web Services to lowercase (if the filename contains uppercasing). This is done to ensure data quality and does not affect the front end display of the content. The 07 Dec 2017 resolved an issue for some articles saved to AWS where the database reference was saved as lowercase, but the filename stored in the XML reference remained as uppercase. The 25 Jan 2018 release performed remediation in the Content Archive in AWS to ensure that packages loaded to Zipline between 10/26/2017 and 12/7/2017 have assets and xml references with matching casing.

Images should be no larger than 1800 pixels wide or high, and no more than 2MB. Silverchair reserves the right to downsample any art larger than these specifications, while ensuring equivalent quality for web viewing. Additional charges may apply if submitted art is consistently outside of these specifications.

There is no lower limit to file size, since quality can vary widely based on image content, but originals should be easily viewed at 100% on screen in a standard browser.