# viaLibri Harvest XML Format

> This document is intended for developers building Harvest implementations from scratch. If you're using WordPress to host your site then we have a number of plugins that will do most of the hard work for you. You can find details at (https://www.vialibri.net/docs/).
>
> If you're building the site in PHP then we have a formatting library that we're considering making available in the composer registry. Let us know if that would be useful and we'll publish it.

In order for your book data to make its way to our database we require your website to provide two XML files that can be accessed via HTTP.

The first XML file contains a) the date and time for the last addition of or update to any book on the site, and b) a list of the IDs of all books currently for sale. We call this the "sync file".

The second XML file contains all the data for every book currently for sale on your site. We call this the "data file".

Once you've implemented both of these and deployed them to your server please let us know the URLs by emailing us at support@vialibri.net.

We will download the sync file every 4 hours. The update date in the XML file is then compared to the books in our database and used to work out whether the data file has any new information since the last sync. We only download the data file if there are new changes to process. The list of IDs in the sync file is then compared to the books in our database to see if there are books we should delete.

A complete sync is performed once a week. When performing a full sync we will always download the data file and will look at data for all books.

## File encoding

Both files should be UTF8 encoded XML. We prefer well-formed XML and recommend using an existing library for generating XML rather than hand-rolling your own solution. However, we try to be quite forgiving in the way we process the files, so they don't necessarily need to pass a strict XML validation test.

It is also not necessary that the file have an .xml file extension. In fact, if the file is not strictly XML compliant we would prefer that it use a different extension. .php, or any other file type that will open in a browser should be OK.

All text fields (including `description`) should be provided as plain text, not HTML. We do make an effort to clean HTML elements from the text in the files, but including them in your file may result in strange characters being displayed in your search listings on viaLibri.

## Securing the files

Most of the time securing the files isn't needed, as all the data that's published is publicly available anyway. However, if you do wish to secure the files then there are a couple of different options:

- Username and password using HTTP Basic authentication.
- Require requests to the URL to have a particular access code in the query string.
- Require requests to the URL to have a particular access code in the Authorization header.

**Do not limit downloads by IP address** as the IP address used will sometimes change.

## The sync file

This file should look like this:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Sync_Data>
  <update>
    <date_update>2016-12-08 19:10:05</date_update>
  </update>
  <ID_set>
    <id>12345</id>
    <id>12346</id>
    <id>12347</id>
    <id>etc...</id>
  </ID_set>
</Sync_Data>
```

The `date_update` field should be in the format yyyy-mm-dd hh:mm:ss and is the date of the most recent insert/update in your database. The timezone doesn't matter and we don't need to know what it is, you just need to be consistent in using the same timezone in this and the data file.

The `ID_set` list contains the database ID of every book that's for sale on the site. This is equivalent to the `source_id` field in the data file. The ordering of this list doesn't matter.

# The data file

```
<?xml version="1.0" encoding="UTF-8" ?>
<Books>
  <Book>
    <date_update>2016-12-08 19:10:05</date_update>
    <author>George Orwell</author>
    <title>Nineteen Eighty-Four</title>
    <description>Nineteen Eighty-Four, often published as 1984, is
a dystopian novel by English author George Orwell published in 1949. The
novel is set in Airstrip One (formerly known as Great Britain), a province
of the superstate Oceania in a world of perpetual war...</description>
    <source_id>12345</source_id>
    <sku_dealer_item_id>ABC123</sku_dealer_item_id>
    <year>1949</year>
    <edition>First edition</edition>
    <publisher>Secker & Warburg</publisher>
    <price>1234.56</price>
    <keywords>dystopian, sci-fi</keywords>
    <isbn>9780547249643</isbn>
    <first_edition>yes</first_edition>
    <signed>no</signed>
    <dust_jacket>yes</dust_jacket>
    <item_url>https://www.example.com/1984/</item_url>
    <image_url>https://www.example.com/1984.jpg</image_url>
  </Book>
  <Book>
    ...
</Books>
```

The Books element contains a Book element for each book currently for sale on the site. Some of the fields in Book are optional, and there are other fields that are not shown here. A full list of the available fields is included below.

The Books should be ordered in decending order of `date_update`, so that the most recently added or updated books are first in the list. You should sort by a secondary value too, to ensure that books with the same `date_update` value are always sorted consistently. Failing to do this may mean that some changes will only be picked up on a full sync.

## Fields

Fields with stars (*) are required.

- **date_update** * - The date that the book's info was last updated. This must by in the format yyyy-mm-dd hh:mm:ss, e.g. 2016-12-16 13:03:12. The timezone doesn't matter and we don't need to know what it is, you just need to be consistent in using the same timezone.

- **author** - The author of the book.
- **title** * - The title of the book.
- **description** - A description of the book. This field should merge, in the preferred order, all data that is needed as part of the book description, such as publisher, condition, place of publication, edition, format, comments, publication date, etc. Data which is included in the `edition` and `publisher` fields will not be displayed unless it has also been added to the field. However, it is possible for our system to automatically add the `publisher` and `year` fields to the start of all your descriptions. Let us know if you'd like us to turn that on for your account.
- **source_id** * - Your database's internal ID for the book.
- **sku_dealer_item_id** * - Dealer's inventory code for the book. This must always be provided, so if you don't use any sort of inventory code just use `source_id` instead.
- **year** - Publication year. This can be just given as four digits, but you may include some text as well. However, any extra text will be ignored.
- **edition** - A description of the edition of the book.
- **publisher** - The book's publisher.
- **price** * - This should not include a currency symbol or any extra formatting. Just the price.
- **keywords** - A set of topics or areas that are relevant for this book.
- **isbn** - The book's ISBN number (if it has one).
- **first_edition** - A yes/no representing whether this book is a first edition or not.
- **signed** - A yes/no representing whether this book is signed or not.
- **dust_jacket** - A yes/no representing whether this book has a dust jacket or not.
- **item_url** * - The full URL for the book on your website. This should be a page that displays the full description for a single book only.
- **image_url** - The full URL for an image of the book. This should be the largest version of the image available. If you don't have an image for this particular book then leave this field blank. *Do not* give us the URL for a placeholder image.

## Formatting boolean values

Boolean yes/no columns such as `first_edition` can be formatted in a number of ways.

- Positive values can be "yes", "y", "true" or "1".
- Negative values can be "no", "n", "false" or "0".

## Paging (optional)

For some sites dynamically generating a file containing data on thousands of books can place a significant burden on the site's server. The time taken to build the file can also result in HTTP requests being terminated as timeouts are reached.

One way to mitigate this is to generate static files periodically and store them on the server. Another way is to break the data file into a number of pages. We will then request only as many pages as are needed to process all the inserts and updates that have been made since

the last sync. This is an optional step, but has the potential to greatly reduce your database load and bandwidth usage.

If you wish to provide your data file in multiple pages then all you need to do is provide the URL of the next page as an extra attribute in your data file. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Books next-page="http://www.example.com/vialibri-data.xml?page=2">
  <Book>
    <date_update>2016-12-08 19:10:05</date_update>
    <author>George Orwell</author>
    ...
```

The number of books included in each page is up to you. The ideal number depends on how powerful your server is. We'd recommend starting with 5000 and testing it out. If a page can be generated and transmitted within 20 seconds then that should be fine.

There are two ways to signal that the last page has been reached. You may use either of these.

1. When returning the last page of results just leave off the `next-page` attribute.
2. If there are no more books to include then simply return a file containing no books. We will stop requesting pages once we've found an empty page.