
Importing and Exporting Data with OJS

Table of Contents

Preface	1
Introduction	1
Articles and Issues XML Plugin	1
Importing Articles and Issues	1
Exporting Articles and Issues From the Web	3
Using the Web-based Expedited Submission Process	4
Users XML Plugin	4
Importing Users	4
Exporting Users From the Web	5
CrossRef XML Export Plugin	5
Exporting Crossref Information From the Web	6
Erudit Article Export Plugin	6
Exporting Erudit Information From the Web	6
PubMed XML Export Plugin	6
Exporting Pubmed Information From the Web	6
Using the Plugins from the Command Line	7
A. Example Articles and Issues Import file	7
B. Example Users Import XML File	8

Preface

Open Journal Systems is a research and development initiative of the Public Knowledge Project at the University of British Columbia. Its continuing development is currently overseen by a partnership among UBC's Public Knowledge Project, the Canadian Center for Studies in Publishing, and the Simon Fraser University Library. For more information, see the Public Knowledge Project web site: <http://pkp.sfu.ca>.

This work is licensed under the Creative Commons Attribution-Share Alike 2.5 Canada License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.5/ca/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Introduction

Open Journal Systems ships with five different Import/Export plugins which allow the Journal Manager to inject data into and extract data from OJS. More information can be found for each plugin's documentation within OJS' plugin directory (`plugins/`).

Import/export functionality is also available as a command-line tool; see `tools/importExport.php` and the command line section for more information.

Articles and Issues XML Plugin

Importing Articles and Issues

OJS features the ability to directly import metadata describing articles and issues for your journal. This can be extremely useful when you are adding back issues to your journal, and do not want to go through the standard OJS editorial process. To do this, you will have to have a basic understanding of XML.

Please note that there is another way to expedite the submission process and publish material quickly which does not depend on making XML files, and may be of use if you only have a handful of items to import. You can view complete instructions here.

Creating the XML Import File

The Articles and Issues XML Plugin is the primary method for batch importing and exporting article-level content. It can be used to import and export single or multiple issues and/or articles, including comprehensive metadata. You can view the complete DTD here: `plugins/importexport/native/native.dtd`. `native.dtd` supports the following as root elements: `<article>`, `<articles>`, `<issue>` and `<issues>`.

You can view a sample XML article import file in appendix A, and also in your OJS installation: `plugins/importexport/native/sample.xml`.

There are a handful of things you should be mindful of when importing data.

- Be sure to define the document type appropriately using `<!DOCTYPE ...>`.
- Your XML file should UTF8-encoded.
- Your XML file must validate. Most *nix systems have access to the `xmllint` [<http://xmlsoft.org/xmllint.html>] program for validation; there are also a number online XML validators, such as Validome [<http://www.validome.org/xml/>]; and many XML editing tools such as Oxygen XML [<http://www.oxygenxml.com/>] and XMLspy [<http://www.altova.com/>] can also validate XML files.
- Dates should be specified as YYYY-MM-DD.
- To import a file, you can use `<embed>` to place a file directly within your XML document, or use `<href>` to link to one.

If you use the `<embed>` tag you will have to base64-encode your files. Using `<embed>` with a base64-encoded file would look something like this:

Example 1. Using the `<embed>` tag to import a file

```
<galley>
  <label>PDF</label>
  <file>
    <embed encoding="base64" filename="my_file.pdf" mime_type="application/
pdf">PGxpPjxCPlc8Zm9ukjfdasD...(and so on)</embed>
  </file>
</galley>
```

You can link to full URLs as well as local files using `<href>`. A full URL link would look like the following:

Example 2. Using the `<href>` tag to import via URL

```
<galley>
  <label>PDF</label>
  <file>
    <href src="http://www.example.com/import_files/my_file.pdf" mime_type="application/
pdf" />
  </file>
</galley>
```

You can use local linking if your galleys are already stored on the destination machine. Importing a local file would look like the following:

Example 3. Using the `<href>` tag to import a local file

```
<galley>
  <label>PDF</label>
  <file>
    <href src="/home/johndoe/import/my_file.pdf"/>
  </file>
</galley>
```

- For importing HTML galleys, use the `<htmlgalley>` element. Otherwise, use the `<galley>` element (ie., for PDF/Postscript, audio, video and other files).
- Some elements can support embedded HTML tags, such as the abstract element. If you do embed HTML within your document, remember to wrap the HTML within `<![CDATA[]]>` tags.
- To add empty data to required fields, you can use the `<![CDATA[]]>` tag as well.

Example 4. Using the `<![CDATA[]]>` tag to add empty data to the required email field

```
<author primary_contact="true">
  <firstname>Jane</firstname>
  <lastname>Doe</lastname>
  <email><![CDATA[ ]]></email>
  <biography locale="en_US">Jane Doe's Bio statement...</biography>
</author>
```

- If you make any typographical errors in the data you are trying to import, you may end up with duplicate or split entries: for example, if your journal already has a section "Articles" with the initials ART, but you mistype in your XML file `<abbrev locale="en_US">AR</abbrev>` instead of `<abbrev locale="en_US">ART</abbrev>`, a new journal section with the initials AR will be created, and that one article will be added to it. This can be easily fixed pre-import, but difficult to clean up after.

Importing Articles and Issues From the Web

The Articles and Issues XML Plugin can be used from the web, from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Articles and Issues XML Plugin** link.
3. Upload your XML file (e.g., `ojsV1N1.xml`).

You will be notified of any errors, or if the import was successful.

Exporting Articles and Issues From the Web

You can export either single or multiple articles and/or issues from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Articles and Issues XML Plugin** link.
3. You will be provided with the options to **Export Issues** or **Export Articles**. Depending on what you choose, you will be able to export either single or multiple articles, or single or multiple issues at a time. These will be exported in XML format, and can be imported to this or another journal at a later date.

Note

Please note that using this plugin to export articles and issues will not only export all relevant metadata, but will include all article files (HTML, PDF, etc.) embedded within the XML document in Base64 encoding. This can result in large, cumbersome XML files, especially when multiple issues are exported at once. Opening them in an editor to view or change any XML data or metadata may be taxing for your computer, and it may take some time to download and/or upload said files, depending on your connection and the resources of the source server.

Using the Web-based Expedited Submission Process

If you only have a handful of files to import quickly, or if you don't want to depend on making XML files to import articles and issues, there is another way to expedite the submissions, available to Journal Managers from the 5-step Author Submission process.

First, make sure you are logged in as a user who is enrolled as both an Editor and an Author. From your User Home, choose the Author Role, and start a submission. Fill out the first two steps of the Author Submission Process as you normally would, keeping in mind that you may need to modify the author metadata information if you are not the original author.

When you reach Step 3 of the submission Process ("Uploading the Submission"), you should upload a galley file, as the file you upload here will appear in the final copyediting queue as both an Initial Copedit file, and as a Galley file. You will still be able to edit, add and remove any galley files at a later step, however. Please note, if you do not upload a file at this point, you will not have the option of expediting the submission process once you are done.

Finish the rest of the submission process as normal. If you have uploaded a file, upon completion of Step 5 you will be presented with the option to move the submission directly to the last step of the Editing Queue. Clicking that link will take you from the submission process to the Editing queue.

You will note that the file you uploaded appears both under Copyediting as an Initial Copyedit file, and under Layout, as a Galley file. For publishing's sake it is the galley file you are interested in. Double-check that the file format has been correctly identified (normally either PDF or HTML), view the proof, and upload any other galley files at this time as well.

You can then schedule the submission under Scheduling, as you would for any other reviewed, copyedited file, and it will appear as soon as the issue it has been assigned to is published. In the case of it being assigned to an already-published issue, it will appear immediately in that issue's Table of Contents.

Users XML Plugin

The Users XML Plugin can be used to import and export users and their roles. See the XML DTD in `plugins/importexport/users/users.dtd` for more information, as well as the sample file in `plugins/importexport/users/sample.xml` and in Appendix B.

Importing Users

You can import single or multiple users from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Users XML Plugin** link.

3. You will find the option to upload an XML User Data file underneath the Export Users section on the Users XML Plugin page.

Creating the Import Users XML File

The Users XML Plugin can be used to import and export single or multiple users. You can view the complete DTD here: `plugins/importexport/users/users.dtd`. The root element is `<users>`.

You can view a sample XML user import file in Appendix B, and also in your OJS installation at `plugins/importexport/users/sample.xml`.

There are a handful of things you should be mindful of when importing user data.

- Be sure to define the document type appropriately using `<!DOCTYPE ...>`.
- Your XML file should UTF8-encoded.
- Your XML file must validate. Most *nix systems have access to the `xmllint` [<http://xmlsoft.org/xmllint.html>] program for validation; there are also a number online XML validators, such as Validome [<http://www.validome.org/xml/>]; and many XML editing tools such as Oxygen XML [<http://www.oxygenxml.com/>] and XMLspy [<http://www.altova.com>] can also validate XML files.
- `<firstname>`, `<lastname>` and `<email>` are mandatory.
- Usernames and passwords are optional, and will be generated automatically if omitted.
- If you allow the system to generate usernames and passwords for the users you are uploading, you can optionally allow the system to email the users with their account credentials. This option can be found on the Users XML Plugin Page ("Send a notification email to each imported user containing the user's username and password.").
- You can require the user change their password when they next log in by setting the password attribute "change" to "true": `<password change="true">myoldpassword</password>`
- The default password encryption is "plaintext" (ie., none). Encrypted assumes it was encrypted by `Validation::encryptCredentials()` and is using the same encryption algorithm used by the system.
- if the imported file contains any usernames or email addresses that already exist in the system, the user data for those users will not be imported and any new roles to be created will be assigned to the existing users.

Exporting Users From the Web

You can export users by one or more roles, or export all users at once. To export one or more roles, simply choose the roles you want exported by clicking (or Ctrl-clicking, if more than one role) the role in the role list and then clicking **Export Users**. To export all users, click the **Export All** link.

Please note that passwords will be encrypted using MD5 or SHA-1 encryption, depending on what system the journal is using.

CrossRef XML Export Plugin

The CrossRef XML Export Plugin provides an export plugin to generate metadata information for Articles and Issues for indexing in CrossRef. Details on the XML format and data requirements is available at <http://www.crossref.org/schema>. More information on this plugin can be found in `plugins/import-export/crossref`.

A valid DOI prefix must be specified in the Journal Setup in order to use this plugin.

The DOI (Digital Object Identifier) Prefix is assigned by CrossRef and is in the format 10.xxxx (e.g. 10.1234).

Exporting Crossref Information From the Web

The Crossref XML Plugin can be used from the web, from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Crossref XML Export Plugin** link.
3. If you have entered your Crossref DOI Prefix in your Journal Setup, you will be provided with the options to **Export Issues** or **Export Articles**. Depending on what you choose, you will be able to export either single or multiple articles, or single or multiple issues at a time. These will be exported in XML format.

Erudit Article Export Plugin

The Erudit Article Export Plugin implements single-article export for full-text indexing using Erudit's XML DTD. See <http://www.erudit.org> for more information.

Exporting Erudit Information From the Web

The Erudit XML Plugin can be used from the web, from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Erudit Article XML Export Plugin** link.
3. You will then be presented with a list of published articles, and links to the XML export for each individual article galley. It is not currently possible to export whole articles or issues.

PubMed XML Export Plugin

The PubMed XML Export Plugin provides an import/export plugin to generate bibliographic information for articles in PubMed standard publisher data format XML for indexing in NLM PubMed/MEDLINE. Details on the XML format and data requirements is available at: <http://www.ncbi.nlm.nih.gov/entrez/query/static/spec.html>. More information on this plugin can be found in `plugins/importexport/pubmed`.

Exporting Pubmed Information From the Web

The Pubmed XML Plugin can be used from the web, from the Journal Manager's User Home:

1. Login as the Journal Manager and from your User Home choose **Import/Export Data**.
2. Then, choose the **Pubmed XML Export Plugin** link.
3. You will be provided with the options to **Export Issues** or **Export Articles**. Depending on what you choose, you will be able to export either single or multiple articles, or single or multiple issues at a time. These will be exported in XML format.

Using the Plugins from the Command Line

Import/export plugins can normally also be run from the command line. You can view the applicable plugin names by running the following command:

```
php tools/importExport.php list
```

You can see usage parameters by running the following command:

```
php tools/importExport.php [PluginName] usage
```

A. Example Articles and Issues Import file

This file can also be found in `plugins/importexport/native/sample.xml`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE issues SYSTEM "native.dtd">
<!--
 * sample.xml
 *
 * Copyright (c) 2003-2007 John Willinsky
 * Distributed under the GNU GPL v2. For full terms see the file docs/COPYING.
 *
 * Sample import document for the OJS 2.x native XML import/export format.
 *
 * $Id: sample.xml,v 1.10 2007/10/10 17:35:09 asmecher Exp $
-->
<issues>
  <issue published="true" current="false" identification="title">
    <title locale="en_US">Test Issue</title>
    <access_date>01-01-2001</access_date>
    <section>
      <title locale="en_US">Articles</title>
      <abbrev locale="en_US">ART</abbrev>
      <article>
        <title locale="en_US">Jane Doe's Article Title</title>
        <abstract locale="en_US">Jane Doe's Article Abstract...</abstract>
        <date_published>2004-10-05</date_published>
        <author primary_contact="true">
          <firstname>Jane</firstname>
          <lastname>Doe</lastname>
          <email>JaneDoe@sample.test</email>
          <biography locale="en_US">Jane Doe's Bio statement...</biography>
        </author>
        <indexing>
          <discipline locale="en_US">Education; Literature Education</discipline>
          <subject locale="en_US">Young adult literature; Holocaust</subject>
          <coverage>
            <geographical locale="en_US">North America</geographical>
            <chronological locale="en_US">Contempoary</chronological>
            <sample locale="en_US">Adolescent readers</sample>
          </coverage>
        </indexing>
      </article>
    </section>
    <htmlgalley locale="en_US">
      <label>HTML</label>
      <file>
        <href src="myfile.html" mime_type="text/html"/>
      </file>
      <stylesheet>
        <href src="myfile.xsl" mime_type="text/xsl"/>
      </stylesheet>
      <image>
```

```
        <embed encoding="base64" filename="myfilename.png"
mime_type="application/png">(base64-encoded data would appear here)
        </embed>
    </image>
    <image>
        <href src="myimage2.png" mime_type="application/png"/>
    </image>
</htmlgalley>
<galley locale="en_US">
    <label>PDF</label>
    <file>
        <href src="mygalley.pdf" mime_type="application/pdf"/>
    </file>
</galley>
</article>
</section>
</issue>
</issues>
```

B. Example Users Import XML File

This file can also be found in `plugins/importexport`.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE users SYSTEM "users.dtd">

<!--
* sample.xml
*
* Copyright (c) 2003-2007 John Willinsky
* Distributed under the GNU GPL v2. For full terms see the file docs/COPYING.
*
* Sample user import data.
*
* $Id: sample.xml,v 1.5 2007/08/30 00:00:06 asmecher Exp $
-->

<users>
    <user>
        <username>testuser1</username>
        <password>ojsajs</password>
        <salutation>Dr.</salutation>
        <first_name>TestUser1FirstName</first_name>
        <last_name>TestUser1LastName</last_name>
        <gender>F</gender>
        <affiliation>TestUser1Affiliation</affiliation>
        <email>testuser1@ojsdev.bb</email>
        <country>DE</country>
        <discipline>C4</discipline>
        <biography locale="de_DE">My biograhay (German) - TestUser1.</biography>
        <biography locale="en_US">My biograhay (English) - TestUser1.</biography>
        <biography locale="es_ES">My biograhay (Spanish) - TestUser1.</biography>
        <role type="author"/>
    </user>
    <user>
        <username>testuser2</username>
        <password encrypted="shal">efe7d096996561e621764052b451ff752b03652f</password>
        <salutation>Dr.</salutation>
        <first_name>TestUser2FirstName</first_name>
        <last_name>TestUser2LastName</last_name>
        <gender>F</gender>
        <affiliation>TestUser2Affiliation</affiliation>
        <email>testuser2@ojsdev.bb</email>
        <country>DE</country>
        <discipline>C4</discipline>
```

Importing and Ex- porting Data with OJS

```
<biography locale="de_DE">My biograhny (German) - TestUser2.</biography>
<biography locale="en_US">My biograhny (English) - TestUser2.</biography>
<biography locale="es_ES">My biograhny (Spanish) - TestUser2.</biography>
<userLocales>de_DE:en_US</userLocales>
<role type="author"/>
  </user>
</users>
```