

FTP deposits to DeepGreen: Guidelines for publishers

DeepGreen Service

Exported on 04/10/2024

Table of Contents

1	Introduction	4
2	General hints to XML	5
2.1	Reserved characters	5
2.2	Unicode (UTF-8)	5
2.3	Namespaces	6
3	ZIP file structure for submission	7
3.1	Multiple articles in a single ZIP file	7
3.2	Deposit examples	7
4	XML metadata format	9
4.1	At publication	9
4.2	Embargo	9
4.3	Licence	10
4.4	Article type / version	12
4.5	Publication date	16
4.6	History dates	17
4.7	Authors / Contributors	17
5	Testing procedure	21
5.1	Publisher account for testing	21
5.2	Test deposits	21
5.3	Validation criteria of DeepGreen	21

Based on "[Publisher protocol for FTP deposits to Publications Router](#)"¹ by the Jisc Open Access Development Team

Version 3.0 (2021-09-15)

¹ https://pubrouter.jisc.ac.uk/static/docs/FTP_deposit_protocol_for_new_publishers.pdf

1 Introduction

This document provides guidelines for generating JATS-XML notifications for submission to DeepGreen (based on Jisc's Publications Router, <https://pubrouter.jisc.ac.uk>). Following carefully these guidelines will enable the notifications to be processed successfully.

In these guidelines information will be given on:

- Notification ZIP file structure for transmission via sFTP
- Metadata to supply for a successful deposit
- DeepGreen testing process.

2 General hints to XML

The Extensible Markup Language (XML) defines a set of rules for encoding documents. Important considerations relating to these rules for the generation of XML are pointed out to in this section in order to make sure that the XML can be processed successfully by DeepGreen.

2.1 Reserved characters

As with every definition of a (computer) language, XML has to make use of special symbols that are reserved, the so-called meta-characters, which cannot be used as content in XML documents.

These include the 5 characters listed in the table below. If you want to send one of these characters you can use any of the codes shown in the table below.

For example, a ">" character would need to be sent within the XML as either `>` or `>` or `>` (the semi-colon is always part of the code!).

Reserved Character	Entity name code	Decical reference code	Hexadecimal reference code
"	<code>&quot;</code>	<code>&#34;</code>	<code>&#x22;</code>
&	<code>&amp;</code>	<code>&#38;</code>	<code>&#x26;</code>
'	<code>&apos;</code>	<code>&#39;</code>	<code>&#x27;</code>
<	<code>&lt;</code>	<code>&#60;</code>	<code>&#x3C;</code>
>	<code>&gt;</code>	<code>&#62;</code>	<code>&#x3E;</code>

Alternative codings for reserved XML characters

2.2 Unicode (UTF-8)

DeepGreen expects that all texts are supplied solely in Unicode (UTF-8) format. Nevertheless, if it should happen that XML content is not UTF-8 encoded, then it will be mandatory to use explicit encoding using Unicode numeric codes. For example, otilde (õ) would be encoded as `õ` and eacute (é) as

é (a comprehensive list of characters and their codes can be found in Wikipedia https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references)

2.3 Namespaces

In case html code is to be included in the body of the JATS-XML file that includes non-JATS namespaces (i.e. element name prefixes), these namespaces have to be defined within the XML.

For example, if it is wanted to make use of an **oasis table element** (expressed as `<oasis:table>`) in the XML, then the corresponding oasis namespace definition has to be added as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<article article-type="review-article" xmlns:oasis="http://www.niso.org/oasis-
exchange/table" >
                                <!--
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA -->
  <front>
    <journal-meta> ...
    <article-meta> ...
    ....
  </front>
  <body>
    <oasis:table>
      <oasis:row>Inclusion criteria</oasis:row>
    </oasis:table>
  </body>
</article>
```

3 ZIP file structure for submission

The mandatory ZIP file structure for sFTP submission to DeepGreen is described by simple rules:

- Notifications for each scientific article should be packaged into **individual ZIP files**.
- **Each notification ZIP file** should include a **single XML document** containing the metadata of the article according to the **NISO JATS-XML** format (i.e. Journal Publishing Tag Set, Version 1.1). (In very exceptional cases, other formats may be negotiated on explicit request only.)
- The ZIP file may also include additional files containing article text and associated artefacts, such as images, in any format (e.g. PDF, WORD, JPG, etc. embedded ZIP files are also allowed)
- DeepGreen strongly recommends that each notification ZIP file has a **flat structure**. However, if you prefer it to have a directory structure, you must ensure that the JATS-XML and PDF files are in the main level.
- The XML file must be valid, i.e. the structure must syntactically be correct and must not contain html characters such as `<` , `>` , `&` , etc.

3.1 Multiple articles in a single ZIP file

We have included the possibility to deposit multiple articles in a single ZIP file (compatible to the output structure of the Atypen publisher system; a.k.a. bulk submission). Hence:

- Notifications for **multiple scientific articles** can be packaged in **one ZIP file**. Enabling, for example, to send an issue of a journal by a single ZIP file.
- For **each scientific article**, the respective content must be placed separately into **different (sub)directories**. Especially, **subfolders** containing a **single XML document** define a single article, i.e. everything in such a folder has to fulfil the requirements that apply to the **individual ZIP** file case as described above.
- **Please note:** Each **(sub)directory** that contains a **single XML document** shall have a **unique name**. Otherwise, it will not be guaranteed that all scientific articles in the bulk are processed correctly (e.g. there might be losses).

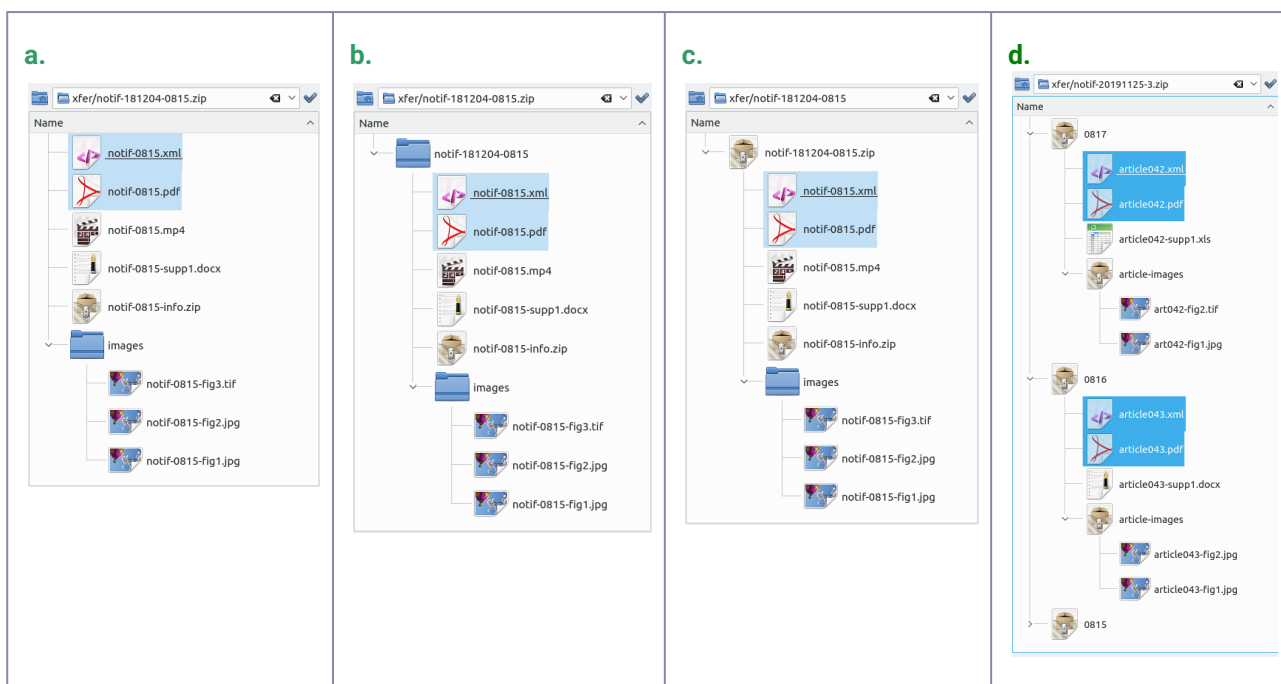
3.2 Deposit examples

All deposits must be done in the `xfer/` folder available to your FTP account on DeepGreen.

Four examples (a. to d.) are shown here:

- a. The notification ZIP file has been saved to the `xfer/` folder and contains PDF and XML files at the top level, along with other files (MP4, DOCX, ZIP) and nested directory (`images/`)
- b. Similar to the first example a., but the PDF, XML and other files are in a nested directory.
- c. The FTP process has created a sub-directory in the `xfer/` folder, and the ZIP file is located in the sub-directory.

- d. The notification ZIP file contains multiple articles, each in a separate folder which follows all the rules that apply to a single article ZIP file (see cases a. to c.)



Result after processing the deposited ZIP file

Regardless of how the notifications are deposited in the `xfer/` folder, DeepGreen will process their contents to produce a `FilesAndJATS.zip` file with a flat structure which includes the following content:

- XML file
- PDF file(s)
- All other, additional files containing everything else from the notification submission

4 XML metadata format

The metadata format of DeepGreen follows primarily NISO JATS specifications for journal publishing (blue tag set).

4.1 At publication

When a scientific work is ready for publication, the appropriate version of the full text should be sent to DeepGreen, together with all relevant metadata, including

- DOI of the article
- Title of the article
- Type of the article ([see 4.4 below \(see page 12\)](#))
- Abstract of the article (only if applicable)
- List of all authors and co-authors, including:
 - Name (i.e. surname and given names)
 - **Affiliation(s), including the name of the institution(s)**
 - ORCID (if possible)
 - Email address(es) (if possible)
- Full bibliographic details:
 - Full publication date (day, month, year)
 - Journal title
 - Journal identifier (such as [e]ISSN)
 - Volume and issue number
 - Page numbers (start and end page or, if applicable, article number)
 - Publisher
- History dates /milestone dates (including **acceptance date**)
- Funding information (if possible, ideally including grant numbers)
- Licensing metadata indicating that the full text attached is available under a creative commons licence (normally CC BY)
- embargo information (if applicable) *for the version of the full text attached to the notification*

4.2 Embargo

IMPORTANT: Please keep in mind that all embargo dates calculated by DeepGreen from the given data are **recommendations only**.

In principle, there are two ways for DeepGreen to establish an embargo for an article:

1. When an **embargo period** has been specified in the database of EZB (German Electronic Journal Library, Regensburg) or SHERPA/RoMEO.

DeepGreen will calculate an embargo end-date by adding the default embargo duration to the publication date. For this reason, it is necessary for a full publication date (day, month and year) to be provided.

- If the publication date is only partially provided (i.e. month and year only), then DeepGreen will calculate the embargo end date using the last day of that month as the publication date.
2. When **licensing information** is specifically provided in the metadata of the article. The embargo end date will be only set when the article has at least one open licence (i.e. any creative commons licence), the earliest of which has a start-date in the future. Then, the earliest start date of any of the given open licences for the article will be taken as the embargo end date. In which case, it will overrule any possible EZB or SHERA/RoMEO database entry.

NOTE: If no valid publication date is provided, then DeepGreen will not be able to reliably establish an embargo end date.

4.3 Licence

The licensing info can be provided in two different ways:

1. Setting a default open access licence in the publisher's account of DeepGreen. This default value will be applied to all deposits where no specific licence information is provided in the deposited metadata of the article. (This would come into effect after any default embargo period, as described above, if you specify one. Otherwise, it would apply immediately.)
2. Providing the licence within the JATS xml file for the appropriate article version (whichever version you have attached to this notification, the one that may be exposed on open repositories). Unless you wish the default licence, described above, to apply, licensing information is mandatory including the licence's URL. Below two options are shown how to provide licensing information correctly in the JATS xml file: (for more information see <https://jats.nlm.nih.gov/publishing/tag-library/1.1/element/permissions.html>).

- a. **Using the basic JATS `<licence>` element** (option 1)

Basic JATS element

```
<permissions>
  <license license-type="open-access" xlink:href="https://
creativecommons.org/licenses/by/2.0/">
  </license>
</permissions>
```

```
<!-- Note: license-type is optional -->
```

A licence text may also be provided by making use of the `<license-p>` element as follows:

```
<permissions>
  <license license-type="open-access" xlink:href="https://
creativecommons.org/licenses/by/2.0/">
    <license-p>This is an open-access article distributed... </
license-p>
  </license>
</permissions>
```

- b. **Adopt the NISO ALI recommendation: Making use of the `<ali:license>` element (option 2)**

Using the element

```
<permissions>
  <license>
    <ali:license_ref xmlns:ali="http://www.niso.org/schemas/ali/
1.0/" start_date="2017-02-03">
      https://creativecommons.org/licenses/by/2.0/
    </ali:license_ref>
  </license>
</permissions>
<!-- Note: start_date is optional; if not provided then licence applies
immediately -->
```

If you want to provide the licence text, then use the `<license-p>` element as follows:

```
<permissions>
  <license>
    <ali:license_ref xmlns:ali=http://www.niso.org/schemas/ali/1.0/
start_date="2017-02-03">
      https://creativecommons.org/licenses/by/2.0/
    </ali:license_ref>
    <license-p>This is an open-access article distributed... </
license-p>
  </license>
</permissions>
```

- c. **Multiple licenses**

If multiple licenses are provided, then a `<license>` element must be present for **each** individual license.

The article's type, however, is a well established attribute, `article-type`, within JATS. It should be filled according a suggested controlled vocabulary as indicated in the table below.

Some realistic article attributes
<pre> <article xmlns:mml="http://www.w3.org/1998/Math/MathML" xmlns:xlink="http://www.w3.org/1999/xlink" dtd-version="1.1" article-type="article" xml:lang="en"> <front>...</front> <body>...</body> <back>...</back> </article> </pre>

Suggested usage of attribute `article-type`

Although designed to accept any text as its value, the following are suggested article identifier types:

<code>article-type</code>	Meaning / Explanation
<code>abstract</code>	The article itself is an abstract (of a paper or presentation) that usually has been presented or published separately.
<code>addendum</code>	A published work that adds additional information or clarification to another work (The somewhat similar value <code>"correction"</code> corrects an error in previously published material.)
<code>announcement</code>	Material announced in the publication (may or may not be directly related to the publication)
<code>article-commentary</code>	A work whose subject or focus is another article or articles; this article comments on the other article(s). (This value would be used when the editors of a publication invite an author with an opposing opinion to comment on a controversial article and then publish the two together. The somewhat similar value <code>"editorial"</code> is reserved for commentary written by an editor or other publication staff.)
<code>book-review</code>	Review or analysis of one or more printed or online books (The similar value <code>"product-review"</code> is used for product analyses.)

article-type	Meaning / Explanation
books-received	Notification that items, for example, books or other works, have been received by a publication for review or other consideration
brief-report	A short and/or rapid announcement of research results
calendar	A list of events
case-report	Case study, case report, or other description of a case
collection	Wrapper article for a series of sub-articles or responses; this value's usage is restricted to articles whose intellectual content appears primarily in <code><sub-article></code> or <code><response></code> .
correction	A modification or correction of previously published material; this is sometimes called 'errata' (The somewhat similar value "addendum" merely adds to previously published material.)
discussion	Invited discussion related to a specific article or issue
dissertation	Thesis or dissertation written as part of the completion of a degree of study
editorial	Opinion piece, policy statement, or general commentary, typically written by staff of the publication (The similar value "article-commentary" is reserved for a commentary on a specific article or articles, which is written by an author with a contrasting position, not an editor or other publication staff.)
in-brief	Summary or teaser of items in the current issue
introduction	An introduction to a publication, or to a series of articles within a publication, etc. ; typically for a special section or issue
letter	Letter to a publication, typically commenting upon a published work

article-type	Meaning / Explanation
meeting-report	Report of a conference, symposium, or meeting
news	News item, normally current but atypically historical
obituary	Announcement of a death, or the appreciation for a colleague who has recently died
oration	Reprint of a speech or oral presentation
partial-retraction	Retraction or disavowal of part(s) of previously published material
product-review	Description, analysis, or review of a product or service, for example, a software package (The similar value "book-review" is used for analyses of books.)
rapid-communication	Fast-breaking research update or other news item
reply	Reply to a letter or commentary, typically by the original author commenting upon the comments
reprint	Reprint of a previously published article
research-article	Article reporting on primary research ; sometimes given short as "article" or "research" though not recommended (The related value "review-article" describes a literature review, research summary, or state-of-the-art article.)
retraction	Retraction or disavowal of previously published material
review-article	Review or state-of-the-art summary article (The related value "research-article" describes original research.)

article-type	Meaning / Explanation
translation	Translation of an article originally produced in a different language

4.5 Publication date

To ensure that DeepGreen is able to capture the publication date correctly, it is important to follow some rules with the `<pub-date>` element:

```
<pub-date date-type="pub">...</pub-date>
```

As shown this element has **always** an attribute `date-type` which should have the value `"pub"`.

NOTE: If the publication date `<pub-date>` does not contain the attribute `date-type` (or, alternatively, the deprecated attribute `pub-type`), it will not be captured by DeepGreen. Consequently, the embargo end date might not be calculated, and it will not be populated in the repositories.

Correct example for the publication date

```
<pub-date publication-format="print" date-type="pub">
  <day>24</day>
  <month>09</month>
  <year>1999</year>
</pub-date>
```

!!! Incorrect example of a publication date !!!

```
<pub-date publication-format="print" date-type="accepted"> <!-- *** WRONG *** -->
  <day>24</day>
  <month>09</month>
  <year>1999</year>
</pub-date>
```

In this case, the `date-type` is not recognised by DeepGreen since any acceptance date is to be sent as a `<history>` date, see next section.

4.6 History dates

Additionally to the publication date, DeepGreen is able to retrieve dates of other publishing workflow events. As long as these are provided as `<history>` XML elements, they will be passed on to repositories.

```
<history>
  <date>...</date>
  <date>...</date>
  ...
</history>
```

Similarly to publication dates, the history `<date>` element **must have an attribute** `date-type` that defines the type of event with one of these values:

- "accepted", "corrected", "preprint", "retracted", "received", "rev-recd", or "rev-request"

Typical history date example

```
<history>
  <date date-type="accepted">
    <day>29</day>
    <month>01</month>
    <year>1999</year>
  </date>
  <date date-type="received">
    <day>05</day>
    <month>05</month>
    <year>1999</year>
  </date>
</history>
```

4.7 Authors / Contributors

Yet another crucial metadata for DeepGreen is the contributor author list. Hence, such a list of authors should be provided by making use of the `<contrib-group>` element as follows:

```
<contrib-group>
  <contrib contrib-type="author">...</contrib>
```

```

<contrib contrib-type="author">...</contrib>
<contrib contrib-type="author">...</contrib>
</contrib-group>

```

A minimum data of **full names and affiliations** are mandatory. Nevertheless, additional metadata such as ORCID or email(s) would also be very useful for institutions. There are different ways in JATS-XML to associate author names with related information. DeepGreen has been designed carefully to capture this information, almost regardless of how it is presented. Some examples are given in what follows.

Basic Example.

Author name, affiliation(s), email, ORCID all located within a particular `<contrib>` element.

```

<contrib-group>
  ...
  <contrib contrib-type="author">
    <name>
      <surname>Cheng</surname>
      <given-names>Yougan</given-names>
    </name>
    <aff>Durham University</aff>
    <email>cheng.yo@durham.ac.uk</email>
    <contrib-id contrib-id-type="orcid">http://orcid.org/0000-0000-0000-0000</
  contrib-id>
  </contrib>
  ...
</contrib-group>

```

Example with affiliation outside of `<contrib>` elements.

Non-cross-referenced affiliations where an `<aff>` (without an attribute `id`) lies within a `<contrib-group>`, but outside of `<contrib>` elements, which DeepGreen automatically applies to all contributors within that group.

```

<contrib-group>
  <contrib contrib-type="author">...</contrib>
  <contrib contrib-type="author">...</contrib>
  ...
  <aff><!-- Affiliation will apply to all contrib authors in the contrib-group --
></aff>
</contrib-group>

```

Examples with cross-references.

Affiliations and emails could also be provided using `<xref>` element, in that case, it is vital to indicate the appropriate `ref-type` attribute.

- **Case 1**

Cross-reference affiliation within the element
<pre> <contrib-group> <contrib contrib-type="author" corresp="yes"> <name> <surname>Shivna</surname> <given-names>Rabindrade</given-names> </name> <contrib-id contrib-id-type="orcid">http://orcid.org/ 0000-0000-0000-0000</contrib-id> <xref ref-type="aff" rid="aff1"></xref> <xref ref-type="corresp" rid="cor1"></xref> </contrib> ... <aff id="aff1">Medical School, University of Michigan, Ann Arbor, MI, USA</ aff> </contrib-group> ... <author-notes> <corresp id="cor1"><email>s.rabin@med.umich.edu</email></corresp> </author-notes> </pre>

- **Case 2**

Cross-referenced affiliation outside of the element
<pre> <contrib-group> <contrib contrib-type="author"> <name>...</name> <xref ref-type="aff" rid="aff99"></xref> </contrib> ... </contrib-group> <aff id="aff99"><!-- Affiliation located outside of a contrib-group --></aff> </pre>

Examples with external links

To satisfy older versions of JATS, email(s) and/or ORCID and can also be provided using the `<ext-link>` element, in this case, it is vital to indicate the `ext-link-type` attribute as "email" or "orcid". It will always be within a `<contrib>` element.

```
<contrib-group>
  <contrib contrib-type="author" corresp="yes">
    <name>
      <surname>Shivna</surname>
      <given-names>Rabindrade</given-names>
    </name>
    <xref ref-type="aff" rid="aff1"></xref>
    <ext-link ext-link-type="orcid">0000-0000-0000-0000</ext-link>
    <ext-link ext-link-type="email">author@email.com</ext-link>
  </contrib>
  <aff id=" aff1">Medical School, University of Michigan, Ann Arbor, MI, USA</aff>
</contrib-group>
```

Note that DeepGreen can successfully process more complex structures than shown above; for example mixing the different approaches outlined above within a single XML file.

5 Testing procedure

In order to join DeepGreen as a publisher, it is strongly recommended to complete a testing and validation process. This involves, among other things, test files being deposited and checked.

5.1 Publisher account for testing

A dedicated user acceptance test (UAT) account will be created by DeepGreen and the sFTP credentials will be sent to the publisher in order to be able to deposit (test) files.

sFTP address	sl61.kobv.de ² (hostname will be subject to change!)
Port	22
User-ID	00000000000000000000000000000000
Password	000000000-0000-0000-0000-00000000

The credentials (user-ID, password) as given in the table are intentionally left unspecified. Each new publisher will obtain a new, individual pair of user-ID and password.

5.2 Test deposits

A number of files need to be deposited, covering different permutations of publications state; these should be as realistic as possible (ideally real publication metadata).

5.3 Validation criteria of DeepGreen

Any test deposits are checked against a comprehensive list of items that must be appear in the XML file within the ZIP file:

- Correct zip file (xml + pdf)
- Valid XML format
- Valid JATS format
- Valid metadata content (depending on publication status)
 - DOI of the article
 - Title of the article

² <http://sl61.kobv.de>

- Type of the article
- Abstract of the article (if provided, i.e. only if applicable)
- List of authors, including the tags
 - Name
 - Affiliation(s)
 - ORCID (if provided)
 - Emails (if provided)
- The full bibliographic details:
 - Full publication date (day, month, year)
 - Journal title
 - Volume and issue
 - Page numbers / article number (if applicable)
 - Publisher
- Keywords (if provided)
- History dates (dates of key publishing milestones / status changes)
- Licensing metadata indicating that the full text attached is available under a creative commons licence (normally CC BY)
- embargo information (if applicable) *for the version of the full text attached to the notification*