

Publisher protocol for FTP deposits to Publications Router

Jisc Open Access Development Team

Document History

Document Owner	Title
Steve Byford	Service Manager

Revision History

Version	Date	Summary of Revision	Revised By
1.0	09-02-2018	Initial Version.	Karmele Hernaiz
1.1	19-04-2018	Licensing info, authors metadata.	Karmele Hernaiz
1.2	11-05-2018	Added details of <ext-link>.	Karmele Hernaiz
1.3	18-06-2018	New section 2 – XML issues	Adam Rehin
1.4	03-07-2018	New section 2.3 - namespace	Karmele Hernaiz
1.5	04-09-2018	Licensing info and new Funding section	Karmele Hernaiz
1.6	07-11-2018	Atypon deposits, publication and accepted date. Document restructure	Karmele Hernaiz
1.7	19-03-2020	Restructured to place priority on deposit of OA VoR at publication for publishers that have signed transformative agreements.	Steve Byford
1.8	29-06-2020	Restructured to include Plan S compliance via green route. Also mandatory etc metadata fields for auto-testing, corresponding author, 1 GB limit.	Steve Byford

Approval

Name	Title	Approval Date
Steve Byford	Service Manager	29-06-2020

Contents

1	Introduction	4
2	XML	4
2.1	Reserved characters	4
2.2	Unicode (utf-8).....	4
2.3	Namespace	5
3	FTP submission	6
3.1	Supported Zip structures.....	6
3.2	Deposit examples.....	7
3.3	Results after processing deposited zip file	8
4	Notification types.....	8
5	Metadata	9
5.1	List of fields required.....	9
5.2	Embargo	9
5.3	Licence.....	9
5.4	Article version	11
5.5	Publication date	12
5.6	Accepted date.....	13
5.7	History Dates	13
5.8	Authors	14
5.9	Funders.....	16
6	Testing process.....	18
6.1	Publisher account for testing.....	18
6.2	Test deposits.....	18
6.3	Router's validation criteria.....	19
7	Addendum: workflow required in the absence of a Plan-S-compliant renewal agreement.....	20
7.1	Notification types	20
7.2	Metadata	20
7.3	Embargoes.....	21

1 Introduction

This document provides guidelines for generating JATS XML notifications for submission to Publications Router, which will enable them to be processed successfully.

It contains information on:

- Notification zip file structure for transmission via FTP
- Publication status changes which should generate notifications
- Metadata to supply
- Router testing process.

2 XML

This section highlights some important considerations relating to the generation of XML so that it may be successfully processed by Router.

2.1 Reserved characters

XML has some symbols that are reserved, which cannot be used as content in XML documents.

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

For example a "<" character would need to be sent within the XML as either **<** or **<** or **<** .

XML Reserved Character Alternative Coding

Reserved Character	Entity name codes	Decimal reference codes	Hexadecimal reference codes
"	"	"	"
&	&	&	&
'	'	'	'
<	<	<	<
>	>	>	>

2.2 Unicode (utf-8)

Router expects all text to be supplied in Unicode (utf-8) format. If you supply XML content that is not utf-8 encoded, then you will need to use explicit encoding using Unicode numeric codes. For example, acute (á) would be encoded as **á** and eacute (é) as **é** (a fuller 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 Namespace

If you need to include html code in the body of the JATS file that includes non-JATS namespaces (i.e. element name prefixes), you should define these namespaces in the XML.

For example, if you want to use an **oasis** table element (expressed as <oasis:table>), then you will need to add the corresponding oasis namespace definition as follows:

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

3 FTP submission

Router accepts submission of article notifications in 2 different Zip file structures. Each of these must be deposited in different directories (folders) on Router's SFTP server.

1. Individual notifications (one per zip file) – these must be deposited in the **xfer** directory
2. Individual or multiple notifications in Atypon zip structure – these must be deposited in the **atypoxfer** directory.

Regardless of the ZIP file structure used, the total size of a *single* notification, when uncompressed, must not exceed 1 GB (1024 MB). Large media files (such as MP4 videos) should be removed from a notification if the size would otherwise exceed 1 GB.

3.1 Supported Zip structures

There are some general rules that apply for both cases:

- **Each notification** should contain a **single XML document** containing the article metadata in **JATS-XML** ([Journal Publishing Tag Set](#)) format
- A notification may also contain 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)
- We recommend that each individual notification 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 both in the main level
- The XML file must be valid. It may contain Unicode characters ($) but not html characters ($) – see section above.

Individual notification deposit

- Each article notification should be packaged into an **individual Zip file**

Atypoxfer zip structure

- A single **Zip file** may have **multiple article notifications**. Each single notification must be contained in individual folders and these, in turn, are grouped together in an issue folder

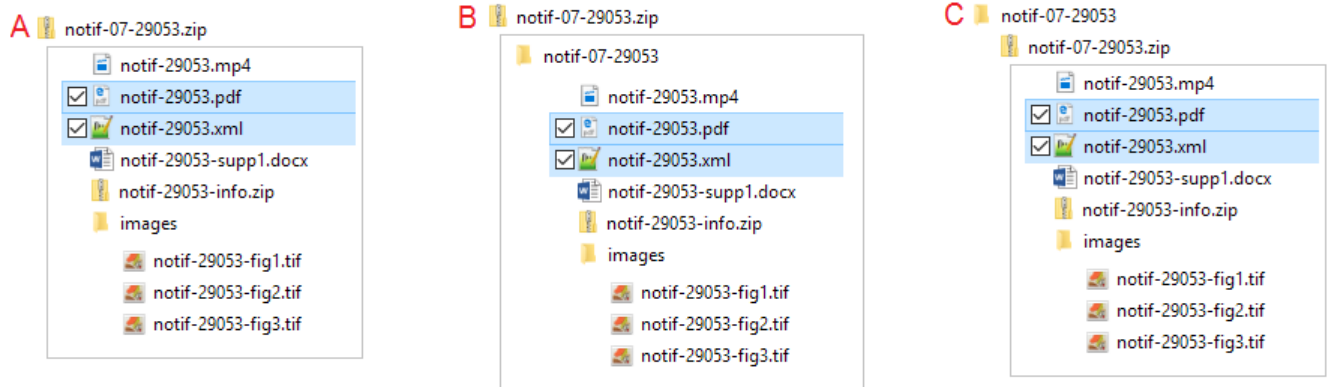
3.2 Deposit examples

Individual notification deposit

All the individual notification deposits must be done in the **xfer** directory available to your FTP account.

Three examples (A to C) are shown here.

- 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)
- Similar to example A, but the PDF, XML and other files are in a nested directory
- The delivery process has created a sub-directory in the **xfer** folder, and the zip file is located in the sub-directory.



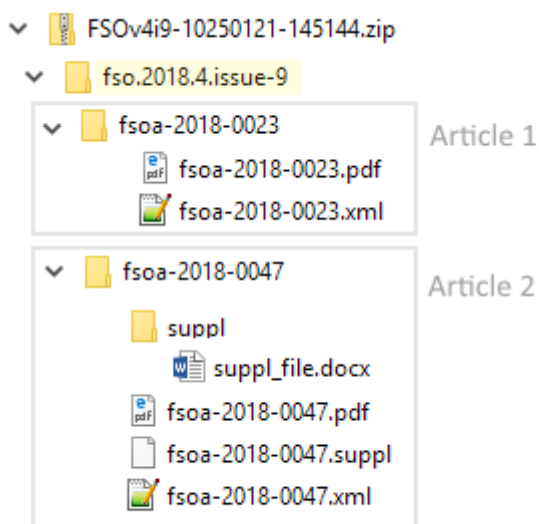
Atypon deposit

All the Atypon deposits must be placed in the **atypon_xfer** directory available to your FTP account.

- Each zip file must contain one issue folder, which includes separate folders for individual notifications.
- Each notification folder must contain PDF and XML files at the top level, along with other files (mp4, docx, zip) and any nested directory.

The image below shows an example of an Atypon zip file with one issue (in highlighted directory) containing two article notifications (in shown in the boxes.)

Example Atypon zip file



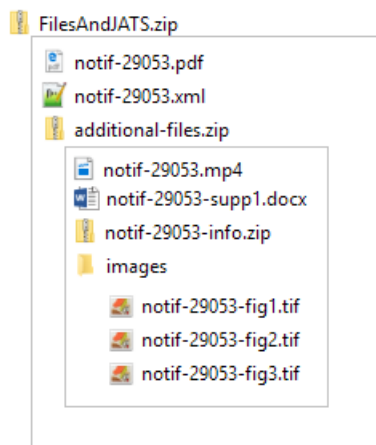
3.3 Results after processing deposited zip file

Regardless of which type of zip file is deposited, Router will process their contents to produce a Router *FilesAndJATS.zip* file which has a particular contents structure:

- XML file
- PDF file(s)
- additional-files.zip file which contains everything else from the notification submission.

Hence the file produced from any of the examples listed above (and sent to Repositories) will have a structure like that shown below.

Example FileAndJATS.zip output file



4 Notification types

Publishers that have signed [transformative agreements](#) with Jisc should provide Router with notifications covering **all OA articles**, including those that have been “flipped” to OA as a result of the agreement, together with those from wholly-OA journals. For each article, please deposit **at the point of publication**. Each notification should contain the complete metadata, together with the full text of the published version of record (VoR).

Publishers that have agreed to enable Plan S compliance via the green route should supply accepted manuscripts (AMs) or the published version of record (VoR) under a CC BY licence and no embargo period (i.e. the CC BY licence should have a start date of no later than the date of publication). *Subject to the final details of UKRI’s OA policy for the post-2021 REF, we are assuming that deposit upon publication will be sufficient.*

For some publishers, we may need a different, two-stage workflow as **detailed at section 7 below**. For publishers that have existing arrangements with Publications Router before the above priorities were in place, we may provisionally accept relaxed rules for mandatory metadata fields.

5 Metadata

This section sets out the article metadata that is required in the XML packages.

5.1 List of fields required

At **publication** stage, the notification should include the article's full text as a PDF, either in the published version of record (VoR) or as an accepted manuscript (AM), together with finalised metadata, including:

- Article version – *mandatory*. Must be one of AM, P, VoR, CVoR, EVoR. See [NISO working group recommendations](#).
- Article's DOI – *mandatory*
- Article's title – *mandatory*
- Article's abstract (if applicable) – *desirable*: please provide if it exists
- Article's type (see [NISO JATS recommendation](#)) – *desirable*
- List all authors and co-authors, including:
 - Name (surname and first name) – *mandatory*
 - Affiliation(s), including the name of the institution(s) – *mandatory*
 - ORCID (if possible) – *highly desirable*
 - Emails (if possible) – *desirable*
- Identify at least one author as the corresponding author – *mandatory*
- The full bibliographic details:
 - Full publication date (day, month, year) – *mandatory*
 - Journal title – *mandatory*
 - Journal identifier – *mandatory* (at least one, such as ISSN)
 - Volume – *mandatory*
 - Issue number – *desirable*
 - Start page number (or article number if applicable) – *mandatory*
 - End page number – *desirable*
 - Publisher – *mandatory*
- History dates / milestone dates: **acceptance date** – *mandatory*; others *desirable*. All dates must be complete (day, month, year).
- Licensing metadata indicating that the full text attached is available under a creative commons licence (normally CC BY) – *mandatory*. There should be no embargo period for Plan-S-compliant solutions as part of renewal deals agreed with Jisc Collections.
- Funding information (if possible, ideally including grant numbers) – *desirable*

5.2 Embargo

Content covered by a Plan-S-compliant agreement with Jisc Collections should have no embargo period. This must be reflected in the article metadata.


If you supply content not covered by such an agreement, the mechanisms for stipulating embargoes are set out in section 7 below.

5.3 Licence

The licensing info can be provided in two different ways:

- 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). **This is the preferred option**

Default open access license (optional)

 [click for more information](#)

License title

License type

License URL

License version

- Setting a default open access licence in your Publications Router account. In that case, it will be applied to all deposits where no specific licence has been included in the deposited article's metadata:

(For content covered by a Plan-S-compliant agreement, there should be no embargo period. For other content, the open licence stipulated here would come into effect after any default embargo period if you specify one as described in section 7.3 below. Otherwise, it will apply immediately.)

Licensing information is mandatory, including the licence's URL. If you include it in the JATS XML file, please follow the format described below (for more information see [JATS specification](#)).

Option 1 – use basic JATS <license> element

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

Note: *license-type* is optional

If you want to provide the licence text, then use 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>
```

Option 2 - adopt the NISO ALI recommendation, use <ali:license> 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>
```

Multiple licences:

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

Correct example:

```
<permissions>
  <license license-type="crossref" xlink:href="http://www.crossref.org/license">
    <license-p> This article is distributed under the terms of the Crossref License </license-p>
  </license>
  <license license-type="open-access" xlink:href="https://creativecommons.org/licenses/by/2.0/">
    <ali:license_ref xmlns:ali=http://www.niso.org/schemas/ali/1.0/ start_date="2018-02-03">
      https://creativecommons.org/licenses/by/2.0/
    </ali:license_ref>
    <license-p>This article is distributed under the terms of the Creative Commons 4.0 License </license-p>
  </license>
</permissions>
```

Incorrect example:

```
<permissions>
  <license license-type="crossref" xlink:href="http://www.crossref.org/license">
    <license-p>This article is distributed under the terms of the Crossref License </license-p>
    <ali:license_ref xmlns:ali=http://www.niso.org/schemas/ali/1.0/ start_date="2018-02-03">
      https://creativecommons.org/licenses/by/2.0/
    </ali:license_ref>
  </license>
</permissions>
```

Note: Do not mix different licence links within the same *<license>* element.

5.4 Article version

Although JATS version 1.1 has no element for directly specifying the article version, it is possible to do so using the *specific-use* attribute of the *<article>* element (as JATS4R recommends), as shown below.

```
<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"
  specific-use="VoR">
...
</article>
```

Note that Router will only capture the article version from specific-use attribute if it has a value defined by NISO JAV (see <https://groups.niso.org/publications/rp/RP-8-2008.pdf>) as shown here:

- AM = Accepted Manuscript
- VoR = Version of Record
- CVoR = Corrected Version of Record
- EVoR = Enhanced Version of Record

The article version indicated as above must be the same as the version in the full-text PDF file supplied with the notification metadata and to which any licensing metadata refers.

JATS version 1.2 and above

JATS v1.2 has an element for supplying the article version information. Although Router does not yet support this, we plan to develop it to do so. More information on this version can be found here:

<https://jats.nlm.nih.gov/publishing/tag-library/1.2d2/index.html>

5.5 Publication date

Publication date must contain the date when the version of record was first published. Although we recommend that only one publication date is provided, it is possible for more than one to be provided with different date-types. In this case we apply rules to select just one: for example, we will take the online publication date in preference to another. (Note that all publication dates are captured and stored as history dates, which are sent to repositories as part of separate article history metadata.)

Due to the importance of the publication date, there are some rules to ensure that Router captures it properly.

JATS provides the **<pub-date>** element for the publication

```
<pub-date>...</pub-date>
```

This element has an attribute: [date-type](#) which should have value of “pub”.

Alternatively, the deprecated attribute [pub-type](#) may be used instead of [date-type](#). It should have one of these values:

- [pub-type](#): epub, ppub, epub-ppub

NOTE: If the <pub-date> publication date does not contain one of these [pub-type](#) or [date-type](#), it will not be captured by Router. Consequently, the embargo end date might not be calculated, and it won't be populated in the repositories.

Correct example:

```
<pub-date publication-format="print" date-type="pub">
  <day>27</day>
  <month>03</month>
  <year>1999</year>
</pub-date>
```

Incorrect example:

```
<pub-date publication-format="print" date-type="accepted">
  <day>27</day>
  <month>03</month>
  <year>1999</year>
</pub-date>
```

(In this case the date-type isn't recognized by Router, which would expect an accepted date to be sent as a <history> date.)

5.6 Accepted date

When providing an article in the acceptance stage, then the accepted date **must** be provided in the <history> element using the *date-type*: "**accepted**" (see example in next section 5.8).

5.7 History Dates

In addition to the publication date, Router will retrieve dates of other publishing workflow events and pass these on to repositories as long as these are provided as <history> XML elements.

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

Similarly to publication dates, the history <date> element must have an attribute that defines the type of event with which it is associated, this must be either:

- [date-type](#) - with one of these values: accepted, corrected, preprint, retracted, received, rev-recd, rev-request
- Or, deprecated, [pub-type](#) – with value: epreprint, ppreprint, ecorrected, pcorrected, eretracted, pretracted.

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>
```

5.8 Authors

Contributing author is also crucial metadata for Router. Therefore, a list of authors should be provided using the **<contrib-group>** element as follows:

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

Please identify the *corresponding* author, by adding a “corresp” attribute to the <contrib> element with the value “yes”.

A minimum data of full names and affiliations are mandatory. However, additional metadata like ORCID or email would be very useful for institutions.

JATS allows various ways to associate author names with related information and Router has been designed to capture this information however it is presented. Some examples are shown below.

Basic Example

Author name, affiliation, 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 *id*) lies within a *<contrib-group>*, but outside of *<contrib>* elements, which Router 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 *ref-type* attribute.

- Cross-reference affiliation within a *<contrib-group>*

```

<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>

```

- Cross-referenced affiliation outside of a <contrib-group>:

```
<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>
```

Examples with external links

To satisfy older versions of JATS, ORCID and emails 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 Router can successfully process more complex structures than shown above; for example mixing the different approaches outlined within a single XML file.

5.9 Funders

Even though the funding information is not mandatory, it is strongly recommended that you provide it as it enables institutional repositories to comply with the [RCUK policy on open access](#). If you include it in the JATS XML file we recommend the first or second format described below (for more information see more: [JATS specification](#)).

Funder and project metadata should be provided using the **<funding-group>**, **<award-group>**, **<funding-source>** and **<award-id>** elements as shown in the 3 examples below. Note that JATS allows alternative constructs for recording funder-Id.

Funder ID provided via <named-content> element

```
<funding-group>
  <award-group>
    <funding-source>
      Funder Name
      <named-content content-type="funder-id">
        https://doi.org/10.10000/000000000000
      </named-content>
    </funding-source>
    <award-id>BR 1528/00</award-id>
    <award-id>SFB 910</award-id>
    <award-id>SFB/TR000</award-id>
    <award-id>SFB/TR444</award-id>
  </award-group>
  <award-group>
    <funding-source>
      Another Funder Name
      <named-content content-type="funder-id">
        https://doi.org/10.11111/111111111111111
      </named-content>
    </funding-source>
    <award-id>PRO-0001</award-id>
  </award-group>
</funding-group>
```

Funder ID provided via xlink:href attribute of <funding-source>

```
<funding-group>
  <award-group>
    <funding-source xlink:href="https://doi.org/10.10000/000000000000">
      Funder Name
    </funding-source>
    <award-id>BR 1528/00</award-id>
  </award-group>
</funding-group>
```

Funder ID provided via <institution-id> element

Note the *institution-id-type* attribute of the <institution-id> element is optional.

```
<funding-group>
  <award-group>
    <funding-source>
      Funder Name
      <institution-id>
        https://doi.org/10.10000/000000000000
      </institution-id>
    </funding-source>
    <award-id>BR 1528/00</award-id>
  </award-group>
  <award-group>
    <funding-source>
      Another Funder Name
      <institution-id institution-id-type="doi">
        https://doi.org/10.11111/1111111111111111
      </institution-id>
    </funding-source>
    <award-id>BR 1528/00</award-id>
  </award-group>
</funding-group>
```

6 Testing process

In order to join Publications Router, it is necessary to complete a testing and validation process which involves test files being deposited and checked. We are currently developing new functionality to automate the process of checking deposited sample files. We will update this document when that functionality is available.

6.1 Publisher account for testing

A UAT account will be created by Jisc and the SFTP credentials will be sent to the Publisher to allow them to deposit files.

SFTP Address	sftp.uat.pubrouter.jisc.ac.uk
Port	22
Username	00000000000000000000000000000000
Password	00000000-0000-0000-0000-00000000

6.2 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).

6.3 Router's validation criteria

- Correct zip file (xml+pdf)
- Valid XML format
- Valid JATS format
- Valid metadata content (depending on publication status)
 - Article's DOI
 - Article's title
 - Article's abstract (if applicable)
 - Article's type (see [NISO JATS recommendation](#))
 - List of authors
 - Name
 - Affiliations
 - ORCID (if possible)
 - Emails (if possible)
 - The full bibliographic details:
 - Full publication date
 - journal title
 - Volume
 - page numbers / article number (if applicable)
 - Publisher
 - History dates (dates of key publishing milestones/ status changes)
 - Licensing and embargo info (if applicable)
 - Funding information (if possible).

7 Addendum: workflow required in the absence of a Plan-S-compliant renewal agreement

7.1 Notification types

If you have not signed a Plan-S-compliant renewal agreement with Jisc, we may ask you to provide Router with notifications for an article at the point of:

- **Acceptance** (or as soon afterwards as possible), when they should contain basic (or provisional) metadata, together with the accepted manuscript.
- **Publication**, containing the updated and complete metadata. Some publishers may feel able to include the full text of the published version of record, perhaps for selected articles; otherwise the deposit should again include the accepted manuscript.

7.2 Metadata

This section documents the details of the article metadata that is required in the XML packages you supply. For publishers that have existing historical arrangements with Publications Router, we may for a while accept relaxed rules for mandatory metadata fields.

At or near acceptance

At **acceptance**, the notification should include basic metadata about the article accompanied by the accepted manuscript (AM) as a PDF file. The metadata should include the following:

- Article version – *mandatory*. Must be AM.
- Article's DOI – *mandatory*.
- Article's title – *mandatory* (even if subject to later change)
- Article's abstract (if applicable)
- Article's type (see [NISO JATS recommendation](#))
- List of all authors and co-authors, including:
 - Name (surname and first name) – *mandatory*
 - Affiliation(s), including the name of the institution(s) – *mandatory*
 - ORCID (if possible) – *highly desirable*
 - Emails (if possible) – *desirable*
 - Identify at least one author as the corresponding author – *mandatory*
- Journal's title – *mandatory*
- Publisher – *mandatory*
- History dates / milestone dates: **acceptance date** – *mandatory*; others *desirable*. All dates must be complete (day, month, year).
- Licensing and embargo info (if applicable) for the AM – *mandatory*. *There should be no embargo period for Plan-S-compliant solutions as part of renewal deals agreed with Jisc Collections. If the embargo end date or post-embargo licence is not yet known, include an explanatory licence text such as "This article is under embargo with an end date yet to be finalized" inside a <p> or <license-p> element within the <license> element. You will then need to include a second open licence (even if provisional, such as CC BY-NC-ND) for the post-embargo period with a start date of e.g. 9999-12-31.*
- Funding information (if possible, ideally including grant numbers) – *desirable*

At publication

At **publication** stage, the notification should include the appropriate version of the full text, together with updated, finalised metadata, including:

- Article version – *mandatory*. Must be one of AM, P, VoR, CVoR, EVoR. See [NISO working group recommendations](#)
- Article's DOI – *mandatory*

- Article's title – *mandatory*
- Article's abstract (if applicable) – *desirable*: please provide if it exists
- Article's type (see [NISO JATS recommendation](#)) – *desirable*
- List all authors and co-authors, including:
 - Name (surname and first name) – *mandatory*
 - Affiliation(s), including the name of the institution(s) – *mandatory*
 - ORCID (if possible) – *highly desirable*
 - Emails (if possible) – *desirable*
 - Identify at least one author as the corresponding author – *mandatory*
- The full bibliographic details:
 - Full publication date (day, month, year) – *mandatory*
 - Journal title – *mandatory*
 - Journal identifier – *mandatory* (at least one, such as ISSN)
 - Volume – *mandatory*
 - Issue number – *desirable*
 - Start page number (or article number if applicable) – *mandatory*
 - End page number – *desirable*
 - Publisher – *mandatory*
- History dates / milestone dates: **acceptance date** – *mandatory*; others *desirable*. All dates must be complete (day, month, year).
- Licensing and embargo info (if applicable) for the version of the full text attached to this notification – *mandatory*. *There should be no embargo period for Plan-S-compliant solutions as part of renewal deals agreed with Jisc Collections*
- Funding information (if possible, ideally including grant numbers) – *desirable*

7.3 Embargoes

Embargoes should only apply to content not covered by a Plan-S-compliant agreement.

There are two ways for Router to establish an embargo for an article:

1. When **licensing** information is specifically provided in the article's metadata.

In this case, an embargo end date will be only set when the article has at least one open 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. (See [Licence](#) section)

- If the embargo end date is not known at the acceptance stage, we suggest using a pessimistic estimate to be revised downwards later, at publication stage. The licence field could then contain a link to a placeholder licence to the effect that “This article is under embargo with an end date yet to be finalized”.

NOTE: “Open licence” refers to any creative commons licence, or any additional licence that the publisher has informed Jisc in advance that it will use as a post-embargo licence.

2. When a **Default embargo period** has been specified in the Router publisher account. In this case, it will be applied to all deposits where no licence is provided in the deposited article's metadata.

Router 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. (See [Publication date](#) section)

- If the publication date is only partially provided (i.e. month and year only), then Router will calculate the embargo end date using the last day of that month as the publication date.

Default embargo period (optional)

 [click for more information](#)

Embargo duration
(months)

[Update embargo](#)

NOTE: if no specific licensing information or valid publication date is provided, then Router will not be able to establish an embargo end date.