6   XML protocol tips and tricks

Version 1.1 - 21/10/2020

Introduction

One of the options available to stakeholders, when it comes to registering new DOIs or updating the information associated to a DOI already assigned, is to use the XML protocol. This protocol is already quite successfully adopted by major genebanks such as NARO in Japan, CGN in the Netherlands or ICARDA in Lebanon and has been used to register about 30% of the total number of DOIs in GLIS.

This document will not go into the details of the XML protocol[1], it will rather focus on the experience gathered assisting early adopters to help future users have an even easier experience.

XML protocol options

The XML protocol is especially suited for large collections, i.e. those for which the batch registration option would be impractical. An important advantage of the XML protocol is that it supports updates while the batch option only supports registration of new DOIs. Three different modalities are available to adopt the XML protocol:

- implementing your own solution
- adopting a Genebank Management Systems (GSM) offering DOI registration/update functions
- deploying the Integration Toolkit

Let's look at each one more in detail.

Implementing your own solution

This is the option that will be discussed in this document. XML is a very popular integration format that, compared to other candidates, has the advantages of being mature, scalable and widely supported by programming languages and environments. Further details on this option will be provided in the following chapters.

Adopting a GLIS-aware GMS

Thanks to the collaboration with CIMMYT, GRIN-Global, a popular Genebank Management System, now offers DOI registration and update functions to make it very easy and convenient to assign DOIs to your collection.

A similar collaboration is being investigated for Web-SDIS, a GMS developed by the SADC Plant Genetic Resource Centre and widely used in southern Africa.

Deploying the Integration Toolkit

The Integration Toolkit is a simple Java application that comes with its own embedded database (it can also use any other JDBC database, if you prefer). The process is as follows:

- you write the information about the PGRFA you want to register or update to the Toolkit's database
- you execute the Toolkit that will prepare the XML messages, send them to GLIS and record the responses in its database

---

[1] See http://www.fao.org/plant-treaty/areas-of-work/global-information-system/techdoc

- you read the responses form the Toolkit database. If a registration was successful, you will get the DOI to record in your own database, otherwise a message error will tell you what went wrong

The Toolkit has been developed in the framework of a Benefit Sharing Fund project in Asia and Africa and has been successfully deployed to over 20 countries in those regions; more than 130,000 DOIs have been registered using the Toolkit so far. If you are interested in checking out the Toolkit, please send an email to PGRFA-Treaty@fao.org and we will be happy to provide you with information, documentation and support should you decide to go ahead.

Getting your credentials

As explained in the XML protocol documentation, GLIS access control for the XML protocol works as follows.

Each system sending registration or update requests to GLIS must use the username/password (system credentials) provided by the GLIS System Administrator. Such username and password are to be indicated in the root element of the XML document, e.g.

```
<?xml version="1.0" encoding="UTF-8">
<register username="xxx" password="yyy">
   <location>
       <pid>DDLLDD</pid>
   </location>
   ...
</register>
```

where "xxx" and "yyy" are the system credentials in question.

If the system credentials are correct, a second check is made on the PID provided in the `location` element that must be associated to the system credentials (as configured by the GLIS System Administrator). This two-stage validation is required because we have systems, such as EURISCO, registering DOIs on behalf of their members. In these cases, a single set of system credentials is associated to EURISCO and all the member PIDs are associated to it.

As the Secretariat maintains a sandbox system where the implementations of the XML protocol, as well as Toolkit deployments, can be tested, you will initially receive a set of credentials and the URL of the sandbox system. When the tests are successfully completed, the GLIS System Administrator will provide you the credentials and the URL for the production server.

Building the XML document

The message that your application will send to GLIS (and the GLIS response to it) will have to follow the XML standard that is widely used for System-to-System integration. In particular, please take note of the following:

- the default character encoding for XML is UTF-8. If you want to use a different character encoding, make sure to indicate it in the XML header, e.g.
  `<?xml version="1.0" encoding="ISO-8859-1">`
  also, make sure that the text content of the XML elements follows the encoding declared in the header!
- the XML document should not be built by just concatenating strings. This is because some escaping may be necessary, depending on the text in your elements. For example, the character "&" is reserved in XML and should be escaped using the corresponding entity "&". Therefore, the Authority "Juz. & Bukasov" should be entered as "Juz. & Bukasov" to avoid XML syntax errors. All programming languages and development environment provide very powerful libraries that automatically take care of the XML escaping requirements as well as character encoding; their use is strongly recommended!

There are very convenient format checkers available online[2] that we recommend you use to check that the XML generated by your application is syntactically correct.

---

[2] One such online service that is quite convenient is https://www.webtoolkitonline.com that, beside validating XML, offers also formatting functions ad also in other formats and languages such as JSON and JavaScript

Sending the request to GLIS

Requests should be sent to GLIS one after the other, i.e.:

1. open the connection,
2. send a request,
3. wait for the response,
4. read the response,
5. close the connection

Only when step 5 above is completed, you can start with the next request. This is necessary to avoid flooding GLIS with too many simultaneous requests. GLIS ensures that only one request at a time is being processed for the same external system. Even with this limitation, GLIS is able to register about 80 DOI per minute or 4,800 per hour; even for large collections, the registration will take just a few hours.

Getting the response from GLIS

GLIS responses are quite simple, especially when the operation was successful. For registrations and updates, the response will be something like:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <sampleid>[sampleID]</sampleid>
  <genus>[genus]</genus>
  <doi>[DOI]</doi>
</response>
```

The presence of the `<doi>` element and the absence of a `<error>` element signal that the operation was successful. What you need to do now is just get the content of the `<doi>` element and record it in your database in correspondence to the `<sampleid>` and `<genus>` indicated in the response.
In case of errors, instead, the response will be like:

```
<?xml version="1.0" encoding="utf-8"?>
<response>
  <sampleid>[sampleID]</sampleid>
  <genus>[genus]</genus>
  <error>Error 1840: Element 'method': [facet 'enumeration'] The value
'Acquisition' is not an element of the set {'acqu', 'ihcp', 'ihva', 'nodi',
'obin', 'obna'}. on line 11
    Error 1824: Element 'method': 'acquisition' is not a valid value of
the local atomic type. on line 11
    Error 1840: Element 'biostatus': [facet 'enumeration'] The value '123'
is not an element of the set {'', '100', '110', '120', '130', '200', '300',
'400', '410', '411', '412', '413', '414', '415', '416', '420', '421', '422',
'423', '500', '600', '999'}. on line 17
    Error 1824: Element 'biostatus': '123' is not a valid value of the
atomic type 'biostatusType'. on line 17
    Error 1839: Element 'historical': [facet 'pattern'] The value 'no' is
not accepted by the pattern '\s{0}|([YyNn])'. on line 19
    Error 1824: Element 'historical': 'No' is not a valid value of the
local atomic type. on line 19</error></error>
</response>
```

In the error response, the is no `<doi>` element and there is a `<error>` element containing the messages of all errors found in validating the message. In the example above, for instance, we had:

- element `<method>` contained the string `Acquisition` instead of one of the allowed codes listed in the error message (`acqu` in this case)
- element `<biostatus>` contained `123` instead of one of the valid values
- element `<historical>` contained `No` instead of a value matching the indicated regular expression (that accepts an empty string or `y` or `n`, possibly in uppercase)

In general, looking at the error message, you will get enough details to be able to quickly fix the message and try again. In case of troubles, please do not hesitate to send an email to PGRFA-Treaty@fao.org and we will assist you.

Another typical error message is the following one, received when attempting to register a PGRFA that has been already given a DOI:

```
<?xml version="1.0" encoding="UTF-8"?>
<response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <sampleid>AzGR-9817</sampleid>
   <genus>Hordeum</genus>
   <error>PGRFA sampleid [AzGR-9817], genus [Hordeum] already
registered for this owner</error>
</response>
```

As you can see, it is clearly indicated that Sample ID AzGR-9817, genus Hordeum, has been already registered for the owner indicated by the PID specified in the `<location>` element.

I am not fully clear on what needs to be done, please help!

Should you require further clarifications, please do not hesitate to send an email to PGRFA-Treaty@fao.org and we will be delighted to assist!