

7 Genesys integration

Version 1.1 - 21/10/2020

Introduction

GLIS is designed to interoperate with external systems such as Genesys¹, EURISCO² and WIEWS³. This document describes the purpose, the workflow and the actual message exchange of the Genesys integration.

Purpose

Genesys is an information portal on Plant Genetic Resources conserved in ex-situ conditions. As many GLIS stakeholders also publish their data on Genesys, it was an easy decision to integrate the two systems. The purposes of the integration are:

1. Add the link to the Genesys page on the specific PGRFA to the GLIS DOI detail page
2. Update GLIS from Genesys information to reduce stakeholder's effort in keeping the two systems up to date

Please note that Genesys does not register DOIs on behalf of stakeholders; they are expected to register DOIs independently using any of the options offered by GLIS. Only when an update is published by the stakeholder to Genesys which contains a GLIS DOI, the integration workflow described in this document takes place.

Workflow

The workflow consists of two phases: the addition to the GLIS DOI page of the link to the Genesys page on the same accession, and the GLIS update from Genesys.

Link addition

Whenever Genesys receives an update from a participating genebank for an accession record, it checks if there is a GLIS DOI associated; GLIS DOIs are identified by the prefix 10.18730/. If there is such a DOI, a message is sent to GLIS requesting the addition of the corresponding Genesys page to the links associated to the DOI.

Requests are grouped up to 500 in the same message for efficiency, also considering that updates to Genesys are usually performed by stakeholders in large batches.

When GLIS receives the request message, for each DOI listed, it:

1. adds the link to the DOI, checking if it is already there, in which case it is updated if necessary
2. queues the request for later processing in the update workflow (see below)
3. returns a response to Genesys about the outcome of the operation

The link is built using a template associated to Genesys, namely:

`https://www.genesys-pgr.org/{doi}`

¹ See <https://www.genesys-pgr.org>

² See <https://eurisco.ipk-gatersleben.de>

³ See <http://www.fao.org/wiews>

where {doi} is replaced with the GLIS DOI in question. This mechanism allows the request message to be as simple as possible, without need of repeating the entire link URL.

GLIS update

When a link addition request is received, as explained above, it is also queued to be reused to query Genesys and obtain the information available in that system to update GLIS.

The queue is consumed by a separate process in GLIS that:

1. fetches the next DOI to be updated
2. queries Genesys to obtain the information about that accession
3. analyzes the JSON document received and compares its content with the GLIS descriptors
4. updates the GLIS descriptors according to the criteria explained below
5. marks the DOI as updated
6. proceeds to the next DOI to update, if any

The WIEWS code of the material holder indicated in the Genesys response must match the GLIS one. If this preliminary validation is passed, the following descriptors are picked from the Genesys response as indicated below

GLIS Descriptor	MCPD	Notes
M02. PGRFA unique identifier	ACCENUMB	
M03. Date	ACQDATE	
M05. Scientific name or crop name	GENUS, CROPNAME	
R03. Biological status	SAMPSTAT	
R04. Additional taxonomic category	SPAUTHOR, SUBTAXA, SUBTAUTHOR	
R05. Names	ACCENAME	
R06. Other identifiers	OTHERNUMB	
R07. MLS status	MLSSTAT	
R08. Historical PGRFA		Genesys historic is used
A01. Provider's location	DONORCODE, DONORNAME	
A02. Provider's PGRFA unique identifier	DONORNUMB	
A03. Country of provenance	ORIGCTY	
A04. Collector's location	COLLCODE	Name, address and country are derived from WIEWS
A05. Collector's PGRFA unique identifier	COLLNUMB	
A06. Collecting mission identifier	COLLMISSID	
A07. Location where the PGRFA was collected	COLLSITE	
A08. Latitude	DECLATITUDE	
A09. Longitude	DECLONGITUDE	
A10. Uncertainty	COORDUNCERT	
A12. Georeferencing method	GEOREFMETH	
A13. Elevation	ELEVATION	
A14. Collecting date	COLLDATE	
A15. Collecting source	COLLSRC	
A16. Breeder's location	BREDCODE	Name, address and country are derived from WIEWS

The MCPD column in the table above corresponds to the Genesys descriptors that are used as data sources.

After the response is analyzed to extract the relevant descriptors and they are normalized to meet GLIS standards (e.g. `historical` is valued 1 or 0 in Genesys and y or n in GLIS), they are compared with the information stored in GLIS.

GLIS descriptors are updated according to the same rules used to validate a registration request. The only differences is that if the Genesys response does not provide any name, other identifiers, collectors or breeders, the ones stored in GLIS are maintained. Otherwise, Genesys names, other identifiers, collectors and breeders are added, avoiding duplication, to GLIS.

Messages

The messages involved in the two workflow phases are described below.

Link addition

When Genesys wants to add its page to the links, the message sent to GLIS is a JSON object as follows:

```
{
  "kws": [ "<k1>", "<k2>", ... ],
  "dois": [ "<doi1>", "<doi2>", ... ]
}
```

where `kws` is an array of the keyword codes taken from the list below and `dois` is the list of DOIs to which the link is to be added with the associated keyword codes. This means that a separate message must be sent for each partition of the keyword codes desired.

Code	Description
1	Passport data
1.1	Genealogy
1.2	Collection documents
2	Characterization
3	Evaluation
3.1	Chemical analysis
3.2	Abiotic stress
3.3	Biotic stress
3.4	Biochemical markers
3.5	Molecular markers
3.6	Cytological characters
3.7	Genomics
3.8	Phenomics
4	Environment
5	Multimedia

Table 1: Keyword codes

For example, to request the addition of a link to DOIs 10.18730/11 and 10.18730/22 with keyword codes 1 and 2 (Passport and Characterization data), the message would be

```
{"kws": [ "1", "2" ], "dois": [ "10.18730/11", "10.18730/22" ]}
```

The response from GLIS is:

```
[
  {
    "doi": "<doi>",
    "result": "<result>",
    "msg": "<msg>"
  },
  ...
]
```

There is one response object in the outer array for each DOI in the incoming request. The content of each object is

`doi` The DOI to which the response object refers
`result` OK or KO depending if the operation was successful or not
`msg` an optional message that provides further details on the outcome of the operation, especially for errors, e.g. a DOI that was not found or an incorrect keyword code

Therefore, a successful response to the previous example would be

```
[
  { "doi": "10.18730/11", "result": "OK" },
  { "doi": "10.18730/22", "result": "OK" }
]
```

In case an error was found for DOI 10.18730/22, assuming it did not exist in GLIS, the response could be

```
[
  {"doi": "10.18730/11", "result": "OK"},
  {"doi": "10.18730/22", "result": "KO", "msg": "No such DOI"}
]
```

GLIS update

As explained above, the link addition request received by GLIS is put in a queue consumed by a separate process. For each DOI in the queue, this process sends a HTTPS GET request to Genesys according to its published API:

```
https://api.genesys-pgr.org/api/v1/acn/{doi}
```

where {doi} is the DOI for which the update is requested.

The Genesys response is a JSON object that is processed as described above, possibly resulting in an update to the DOI record.