The International Treaty
ON PLANT GENETIC RESOURCES FOR FOOD AND AGRICULTURE

Global Information System of the ITPGRFA

Integration Toolkit

Installation and operation manual

Version 0.1 - 25/06/2017 Marco Marsella

History of changes

Version	Date	Description
0.1	25/06/2017	Initial version

Introduction

The Global Information System (GLIS) of the International Treaty on Plant Genetic Resources for Food and Agriculture (ITPGRFA), as described in Article 17 of the ITPGRFA, aims at facilitating "the exchange of information, based on existing systems, on scientific, technical and environmental matters related to plant genetic resources for food and agriculture". The cornerstone of GLIS are information on Plant Genetic Resources for Food and Agriculture (PGRFAs) as they are made available on the Web.

To facilitate finding and accessing such information, GLIS offers the assignation of Digital Object Identifiers (DOIs) to PGRFAs and the collection of links to information resources (targets) available on the Web. The registration of PGRFAs in GLIS to obtain the corresponding DOIs is obtained, among other options, through a XML-based protocol described at

http://www.fao.org/plant-treaty/areas-of-work/global-information-system/techdoc/en/

To promote such XML protocol, the Treaty Secretariat has developed an Integration Toolkit (or Toolkit in short) providing the necessary XML formatting and communication layer thus greatly simplifying the operation for adopting stakeholders.

Adoption of the Toolkit is not required to participate in the GLIS initiative, it is offered as an alternative to those stakeholders that are unable or unwilling to implement the XML protocol but have a large enough collection to make the other available options impractical.

Future versions will likely extend the features of the Toolkit to cover other services beyond those described in this document.

For any clarification, support request or to contact the Treaty Secretariat, please send an email to PGRFA-Treaty@fao.org.

System prerequisites

At this moment, the Toolkit only works under Linux with MySQL as database backend. The detailed system requirements are:

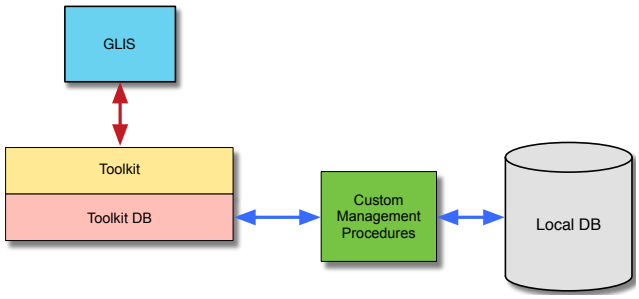
- Linux, any recent distribution such as Ubuntu 16.04 or later
- Oracle Java 8+ with JAVA_HOME properly set
- MySQL 5.7 or later with the corresponding JDBC driver
- 8GB RAM (16GB preferred)
- 64GB hard disk or more, depending on the size of your collection

The Secretariat is available to consider supporting other configurations on request.

Adopting the Toolkit

The Toolkit can be seen as a middleware taking information from its own database, converting it to the corresponding XML message, sending it to GLIS and recording the result again to the database. The rationale of this architecture being that it is more likely to find enough expertise in interested stakeholders to read and write data to the Toolkit database than it would be to implement a robust XML communication layer.

The resulting architecture is described in the following diagram.



Adopting the Toolkit requires the implementation of the "Custom Management Procedures" box. This component will essentially contain the logic required to extract information from the local database mapping its structure to that of the Toolkit DB for the descriptors that need to be sent to and received from GLIS. This activity will need to take into account the documents available at:

http://www.fao.org/plant-treaty/areas-of-work/global-information-system/guidelines/en/

http://www.fao.org/plant-treaty/areas-of-work/global-information-system/descriptors/en/

for, respectively, use cases for the assignation of DOIs and the descriptors involved. Although it is recommended that a suitable application is developed to implement the "Custom Management Procedures", it is possible to devise a set of database operations exporting the necessary data from the local DB and writing back the result of the operation returned by GLIS .

The Toolkit is supposed to operate as a black box reducing the burden on the stakeholder being guaranteed that, if proper information is provided to the Toolkit DB, the transaction with GLIS will be successful and that future development of the GLIS workflow will be supported by the corresponding release of the Toolkit.

The Toolkit database

The Toolkit database is used to exchange data to and from your local database. Essentially, you write data to the Toolkit database, execute the Toolkit and check the result of such operation in the Toolkit database.

The Toolkit is indeed a processor of the XML messages described in the "XML Integration Protocol" document available at

http://www.fao.org/plant-treaty/areas-of-work/global-information-system/techdoc/en/

The content of the database is very closely related to the XML elements described in the document with some exceptions described below. Otherwise, please refer to the XML element indicated next to table columns, when applicable.

The Toolkit database includes the following tables:

actors

Stores information about providers, collectors and breeders associated to the PGRFA. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. Any number of collectors and breeders can be present, but only one provider. The columns are as follows:

Column	Description
`id`	Primary key
`pgrfa_id`	Foreign key to `pgrfas.id`
`role`	A 2-letter code identifying the actor role. `pr`: provider, `co`: collector and `br`: breeder
`wiews`	The FAO/WIES Institute code, if available
`pid`	The Easy-SMTA PID, if available
`name`	The organization name or the name of the individual
`address`	The organization address of the individual address
`country`	The ISO-3 code of the organization country or the individual country

identifiers

Stores information about additional identifiers not already provided in the `pgrfas` table. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the

transfer transaction to receive the descriptors associated to the incoming PGRFAs. Any number of identifiers can be present. The columns are as follows:

Column	Description
id	Primary key
pgrfa_id	Foreign key to pgrfas.id
type	The code of the identifier type. See Table 4 of the XML Integration protocol document
value	The identifier value

names

Stores names associated to the PGRFA. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. Any number of names can be present. The columns are as follows:

Column	Description
id	Primary key
pgrfa_id	Foreign key to pgrfas.id
name_type	A 2-letter code of the type of the name. cn: crop name, on: other name
name	The identifier value

pgrfas

This is the main table and stores information associated to the PGRFA. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. In this case, the columns refer to the descriptors provided to GLIS upon the PGRFA registration performed by the provider. The columns are as follows:

Column	Description
id	Primary key
operation	The operation associated to the PGRFA. Values recognized by the Toolkit are: register, update or transfer (for incoming material. Outgoing transfers are managed through a separate table, see below)
sample_id	The sample identifier in your collection or the identifier in the provider's collection
processed	y or n indicating whether the row has been processed, regardless of the outcome, see the results table. For incoming transfers y means that the row is ready to be read to copy the descriptors to your local database.
sample_doi	The DOI associated to the PGRFA. The role of this value depends on the operation as follows: register leave empty if you want GLIS to assign a DOI to the PGRFA. Enter here the existing DOI you obtained using some other service and are registering the PGRFA to GLIS update must contain the DOI (minted by GLIS or not) associated to the PGRFA that must be already registered in GLIS transfer for incoming transfers, this is the DOI associated to the provider's PGRFA that you are receiving. You should provide it as the progenitor's DOI (see table progdois below) when registering the PGRFA in GLIS once it is incorporated into your collection. This is critical to establish the correct relationship between the provider's DOI and the DOI of the PGRFA your received.
date	See XML Integration protocol [date] M03
hold_wiews	See XML Integration protocol [lwiews] M01
hold_pid	See XML Integration protocol [lpid] M01
hold_name	See XML Integration protocol [lname] M01
hold_address	See XML Integration protocol [laddress] M01
hold_country	See XML Integration protocol [lcountry] M01
method	See XML Integration protocol [method] M04
genus	See XML Integration protocol [genus] M05

species	See XML Integration protocol [species] R04
sp_auth	See XML Integration protocol [spauth] R04
subtaxa	See XML Integration protocol [subtaxa] R04
st_auth	See XML Integration protocol [stauth] R04
bio_status	See XML Integration protocol [biostatus] R03
mls_status	See XML Integration protocol [mlsstatus] R07
prov_sid	See XML Integration protocol [psampleid] A02
provenance	See XML Integration protocol [provenance] A03
coll_sid	See XML Integration protocol [csampleid] A05
coll_miss_id	See XML Integration protocol [missid] A06
coll_site	See XML Integration protocol [site] A06
coll_lat	See XML Integration protocol [clat] A06
coll_lon	See XML Integration protocol [clon] A06
coll_uncert	See XML Integration protocol [uncert] A10
coll_datum	See XML Integration protocol [datum] A11
coll_georef	See XML Integration protocol [georef] A12
coll_elevation	See XML Integration protocol [elevation] A13
coll_date	See XML Integration protocol [cdate] A14
coll_source	See XML Integration protocol [source] A15
ancestry	See XML Integration protocol [ancestry] A17

progdois

Stores the DOI(s) of the progenitor(s) associated to the PGRFA. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. The number of DOIs that can be present depends on the method; please see the XML Integration protocol document for details. The columns are as follows:

Column	Description
id	Primary key
pgrfa_id	Foreign key to pgrfas.id
doi	The progenitor's DOI. Must be registered in GLIS

results

This table stores the result of the operation applied to the corresponding row of pgrfas. The columns are as follows:

Column	Description
id	Primary key
operation	The operation that was requested (see pgrfas above)
genus	The genus of the PGRFA
sample_id	The sample ID passed in the column pgrfas.sample_id
doi	The DOI associated to the PGRFA. For registration operations, this is the newly minted DOI associated to the PGRFA
result	OK or KO depending whether the operation was successful or not
error	The error message returned by GLIS, if any

settings

This table stores some configuration parameters required by the Toolkit. The columns are as follows:

Column	Description
setting_name	The name of the setting
setting_value	The value of the setting

The following settings are defined:

Setting name	Description
`username`	The username to be used in the registration, update or transfer requests being sent to GLIS. Must be provided by the GLIS System Administrator
`password`	The password to be used in the registration, update or transfer requests being sent to GLIS. Must be provided by the GLIS System Administrator

Additional settings may be defined in further versions of the Toolkit.

targets

Stores the targets, i.e. links to associated information on the PGRFA available on the Internet. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. The columns are as follows:

Column	Description
`id`	Primary key
`pgrfa_id`	Foreign key to `pgrfas.id`
`value`	The target value (usually the URL to the information resource associated to the PGRFA)

tkws

Stores the target keyword codes, i.e. the codes corresponding to the keywords associated to each target. It is used for registration of PGRFAs to GLIS and update of already registered PGRFAs. It is also used in the transfer transaction to receive the descriptors associated to the incoming PGRFAs. The columns are as follows:

Column	Description
`id`	Primary key
`target_id`	Foreign key to `targets.id`
`value`	The keyword code. Please see XML Integration Protocol table 2

transfers

Stores the information about the transfer being notified to GLIS. The columns are as follows:

Column	Description
`id`	Primary key
`symbol`	The identifier assigned by the Provider to the transfer. For SMTA transfers, this is the SMTA symbol
`processed`	`y` or `n` indicating whether the row has been processed, regardless of the outcome, see the `transfer_results` table
`providerpid`	The Easy-SMTA PID of the Provider
`recipientpid`	The Easy-SMTA PID of the Recipient
`type`	The type of the transfer (`SMTA` or `MTA`)

transfer_materials

Stores the DOIs of the material being transferred, one row for each material. The columns are as follows:

Column	Description
`id`	Primary key
`transfer_id`	Foreign key to `transfers.id`
`doi`	The DOI associated to the material being transferred

transfer_results

Stores the result of the transfer. The columns are as follows:

Column	Description
id	Primary key
result	OK or KO depending whether the operation was successful or not
providerpid	The Easy-SMTA PID of the Provider
recipientpid	The Easy-SMTA PID of the Recipient
symbol	The identifier assigned by the Provider to the transfer. For SMTA transfers, this is the SMTA symbol
type	The type of the transfer (SMTA or MTA)
error	The error message returned by GLIS, if any

Getting the Toolkit

The Toolkit is a WSO2 application that is distributed as a compressed archive. To obtain it, please contact pgrfa-treaty@fao.org. You will get a file named GLIS-Toolkit-x.x.x.zip where x.x.x is the version number (currently 1.0.0). Inside the archive, you will find documents providing detailed instructions on how to deploy the Toolkit and all necessary files required to do so.

Starting WSO2

To start WSO2, type the following commands in the sequence shown and from two separate terminal windows:

From terminal window 1: wso2_home/bin/broker.sh

From terminal window 2: wso2_home/bin/integrator.sh

You may want to configure WSO2 as a service. Please refer to the WSO2 documentation for details. Please note that starting WSO2 as a service implies the execution of components of the WSO2 suite that are currently not being used increasing the RAM requirement for the server.

Stopping WSO2

To stop WSO2, if you started it from terminal windows as described above, just press CTRL-C in the Integrator terminal window and then again in the Message Broker window. If you installed WSO2 as a service, please issue the corresponding command found in the WSO2 documentation.

The WSO2 web interface

WSO2 offers a web-based management interface that is accessible through any modern browser. The web pages of interest to us are:

https://<tk_host>:8243/carbon/ for the Integrator

https://<tk_host>:9446/carbon/ for the Message Broker

The access credentials in both pages are admin/admin. To change them, please refer to the WSO2 documentation for details.

Using the Toolkit

Once the WSO2 is running and the Toolkit has been deployed, it is possible to use its functions to

- register new PGRFA to GLIS materials and obtain the newly assigned DOI,
- update the descriptors associated to PGRFA already registered to GLIS, and
- inform GLIS of PGRFA transfers from your institution to a recipient

Please note that the transfer transaction only works if the recipient also deployed the Toolkit or provides a similar service able to understand the messages that GLIS will send to it. See the chapter about the transfer transaction below for details.

The Toolkit uses its own database to read and write the information it needs or obtains during its operations, as explained in the diagram provided at the beginning of this document.

To use the Toolkit, you will need to write a set of management procedures reading information from your own database and writing them to the Toolkit's DB and vice versa, depending on the function that

you intend to use. The Toolkit database tables are used as input to the Toolkit and as output from the Toolkit.

In the following chapters, details are provided for the three functions listed above.

The Toolkit trigger

To invoke the execution of the Toolkit functions, you can type the following command from the `toolkit-home` directory

```
java -jar glis-trigger.jar
```

from a new terminal window.

The trigger will extract the information from the Toolkit's database and invoke the Toolkit's API endpoints to perform the required functions. It is possible to execute the trigger as a `cron` job running periodically, say every 5 minutes, but this would require careful population of the Toolkit's database tables to avoid undesired processing of partially populated rows.

The trigger is a simple java application designed to:

1. Check if the file `lock.lck` is present in the same directory as `toolkit-home`. If yes, this means that another instance of the trigger may be running. To avoid conflicts, emit an error message and exit. If not, continue to step 2 below
2. Create a file `lock.lck` in the same directory as glis-trigger.jar
3. Select all rows of table `pgrfas` with `toolkit-home`
4. Order them by increasing `id` and limit the result set to a maximum of 1.000 rows
5. Fetch the next `pgrfas.id` from the result set
6. Invoke the Toolkit API service entry point

   http://glistk:8280/GLIS_Registration/<id>

   where <id> is the `pgrfas.id` mentioned above
7. If the message is correctly received by the Toolkit, mark the `pgrfas` row as processed, setting `pgrfas.processed = 'y'`. This is because, thanks to the Guaranteed Delivery of messages offered by the Message Broker, it is assumed that the message will be eventually sent to GLIS
8. Repeat steps 3, 4 and 5 above until the result set is exhausted
9. Delete the file `lock.lck`
10. Exit

If, for any reason, the trigger is interrupted, the file `lock.lck` will be left in the same directory as glis-trigger.jar. In this case, attempts to run the trigger again will result in the following error being displayed:

```
Trigger already running, EXIT
```

To resume normal operation, make sure there is no other instance of the trigger actually running (be particularly careful if you configured `cron` to run the trigger periodically) and just delete the `lock.lck` file.

The trigger can be replaced by a function in your own "Custom Management Procedures" if this would be preferable. In this case, your own application will invoke the Toolkit API entry point as required.

The results table

When the Toolkit executes an operation (e.g. the registration of a new PGRFA), it stores the result of such operation into the `results` table. Your management procedure will look at the `results` table to find out if the operation was successful and collect any output provided (in our example, the DOI assigned by GLIS). Should the operation have failed, the `results` table will provide an error message that can be used to resolve the issue. Please note that the `processed` column in the `pgrfas` table is set to `y` by the trigger regardless of the outcome of the operation. Therefore, if an error occurred for a specific PGRFA, you should:

- Look at the error message in the results table
- Perform any correcting action to eliminate the problem
- Delete the row from the results table
- Set the processed column of the corresponding record in `pgrfas` to `n`
- Invoke the trigger to try the operation again with the fix

The housekeeping of the `results` table is your responsibility; when you read a row from the `results` table, you should remove it.

The results table contains the sample_id and the genus (if provided in the request) to help you identify the proper PGRFA record in your database. This because, in some institutions, the sample_id alone is not sufficient to uniquely identify the PGRFA to which the result applies.

PGRFA registration to obtain the corresponding DOI

The purpose of this function is to register a PGRFA to GLIS and obtain the newly assigned DOI. To do so, you will populate some of the Toolkit's tables with descriptors extracted from your database.

The tables in the Toolkit's database involved in this function are:

pgrfas — This is the base table containing one row for each PGRFA to be processed. Populate the row with the descriptors coming from your local database. Make sure that processed is set to n until all corresponding rows in related tables are completely populated. For this function, the column sample_doi must be left NULL.

actors — This table stores information on the provider, collector(s) and breeder(s) associated to the pgrfas row. The pgrfa_id column is the foreign key to the column pgrfas.id. The role column is as follows:

pr	Provider
co	Collector
br	Breeder

For any given PGRFA, there must be zero or one provider; instead, zero or more collectors and zero or more breeders can be provided.

identifiers — This table stores the other identifiers associated to the PGRFA

names — This table stores the crop (cn) names and the other (on) names

progdois — This table stores the progenitor(s) DOIs. Such DOIs must be already registered to GLIS

targets — This table stores the URLs of web resources about the PGRFA

tkws — This table stores the keyword codes associated to each target pointing to targets.id as foreign key

results — This table stores the result of the registration request

Once the tables are populated with the information provided, invoke the trigger (or wait for it to be invoked by cron) and look at the results table to get the DOI associated to each PGFRA or identify what went wrong.

Update of descriptors associated with a PGRFA

The purpose of this function is to update one or more descriptors associated with a PGRFA already registered to GLIS. To do so, you will populate some of the Toolkit's tables with descriptors extracted from your database. You must provide all applicable descriptors, not just the changed ones, because GLIS replaces the entire PGRFA record with what you provide.

The tables in the Toolkit's database involved in this function are:

pgrfas — This is the base table containing one row for each PGRFA to be processed. Populate the row with the descriptors coming from your local database. Make sure that processed is set to n until all corresponding rows in related tables are completely populated. For this function, the column sample_doi must be populated with the DOI obtained from the registration function described above. sample_doi is used to uniquely identify the PGRFA record in GLIS that you want to update.

actors — This table stores information on the provider, collector(s) and breeder(s) associated to the pgrfas row. The pgrfa_id column is the foreign key to the column pgrfas.id. The role column is as follows:

pr	Provider
co	Collector
br	Breeder

For any given PGRFA, there must be zero or one provider; instead, zero or more collectors and zero or more breeders can be provided.

`identifiers`	This table stores the other identifiers associated to the PGRFA
`names`	This table stores the crop (`cn`) names and the other (`on`) names
`progdois`	This table stores the progenitor(s) DOIs. Such DOIs must be already registered to GLIS
`targets`	This table stores the URLs of web resources about the PGRFA
`tkws`	This table stores the keyword codes associated to each target pointing to `targets.id` as foreign key
`results`	This table stores the result of the registration request

Once the tables are populated with the information provided, invoke the trigger (or wait for it to be invoked by `cron`) and look at the results table to check whether the update was successful or identify what went wrong.

PGRFA transfer from provider to recipient

The purpose of this function is to inform GLIS of the transfer of PGRFAs from a provider to a recipient. The result of a successful transaction will be the accurate transfer of information associated to the DOI of the provider's PGRFAs to the recipient's system and, if so chosen by the recipient, the acquisition of new DOIs associated to the received PGRFAs once incorporated in the recipient's collection and proper relationship being established between the provider's DOI and the recipient's DOI for any material incorporated into the recipient's collection.

Prerequisites

In order for the transaction to work, it is necessary that:

- The provider is registered in Easy-SMTA (www.planttreaty.org/mls) and has obtained the corresponding PID
- The recipient is also registered in Easy-SMTA and has obtained the corresponding PID
- All the PGRFAs being exchanged have already been assigned a DOI (from GLIS or other services) and have been registered in GLIS
- The recipient has an instance of the Toolkit running on his system or an equivalent service supports the transfer transaction described in the XML Integration protocol document
- The recipient has provided the Treaty Secretariat with the URL of the Toolkit (or equivalent service) to be recorded in Easy-SMTA under the field Toolkit URL available in the user profile

It is critical that all of the above steps have been performed accurately for the transaction to be successful.

Provider actions

To initiate a transfer request, the Provider will:

1) Create a new rom in the `transfers` table with:
    a. Its own PID in column `providerpid`
    b. The Recipient's PID in column `recipientpid`
    c. The symbol of the SMTA or the identifier of the MTA regulating the transfer in column `symbol`
    d. The type of the transfer (`SMTA` or `MTA`) in column `type`
    e. Set column `processed = 'n'`
2) Create one or more new rows in the `transfer_materials` table with:
    a. The id of the newly created row of transfers in column `transfer_id`
    b. The DOI of the material being transferred in column `doi`
3) Invoke the trigger or wait for its next execution if installed as a `cron` job.

The trigger will find the transfer row(s) with `processed='n'` and invoke the appropriate Toolkit entry point. In turn, the Toolkit will assemble a suitable XML message and send it to GLIS.

Upon receipt of the transfer request, GLIS will check the validity of the information provided and, if the Recipient has provided the Toolkit URL, forward the request to the Recipient's Toolkit (or equivalent service). Finally, a positive acknowledge will be returned to the Provider's Toolkit to be recorded in the `transfer_results` table.

Should an error be found by GLIS, or should the forwarding of the request to the Recipient's URL fail, a negative acknowledge will be sent back to the Provider's system to be recorded in the `transfer_results` table for later inspection.

Recipient actions

Assuming that the transfer request is found to be valid by GLIS and that it is forwarded successfully to the Recipient's Toolkit, this latter system will:

1) Archive the request message for later reference, if required
2) For each DOI listed in the `<materials>` element, query GLIS to obtain the associated information and store it in the Toolkit database tables. Note that the new row in the `pgrfas` table will be marked with `operation='transfer'` and `processed='y'` for the custom management procedures to easily identify the new incoming PGRFAs

At this time, the information associated to the PGRFAs received from the Provider is available in the various tables of the Toolkit DB. Please note that the query to GLIS returns all information *except* the relationships to other PGRFAs. It will now be the responsibility of the custom management procedures to extract the information from the Toolkit database and make it available to the local database.

It is very important to note that the `pgrfas.sampledoi` column will contain the DOI that the Provider assigned to the PGRFA being transferred. This information is very important and should be maintained for later use (see below). If storing this information in the local database is not advisable, it should be left in the Toolkit database.

After the necessary procedures have been completed by the Recipient organization and the PGRFAs are actually incorporated into the Recipient's collection (this may take quite a long time, depending on the individual PGRFAs, the steps involved, the regulations and so on), it is possible to assign new DOIs to the newly acquired material using the registration transaction described above. In this case, it is strongly recommended to include the Provider's DOI of the PGRFA (see above) in the `progdois` table. This will allow GLIS to establish a relationship between the Provider's DOI and the Recipient's DOI thus allowing users to navigate through the chain of relationships among the associated PGRFAs.