UI-less Data Submission Protocol

Version1.8 February 17, 2016

# Overview

This document describes bulk automated data submission protocol for NCBI archives.

To submit data, submitter needs to prepare a submission package that consists of:

1. Data files (actual data files)
2. Submission XML (description of what need to be done with these files, plus metadata)

Each submission package contains set of actions. Each action is associated with a single target database (NCBI archive, such as BioSample, BioProject, GenBank WGS, etc.). Processing of each action results in creation of a single “object” in target database.

Created object could be aggregate, i.e. contain multiple parts with many accessions. For instance WGS processing will create genome that contains multiple contigs and “WGS master”. But the action processing is atomic, it either completely succeed or completely fails. In case of failure no partial objects are created and the action has to be re-submitted as a whole.

DB1

Submission

Action

Action

DB2

Upon successful processing of initial submission, Submission Portal returns an NCBI Accession for the created object. In case of aggregate object, single accession is returned. Usually this accession points to a record linking together all parts of the submission. For instance for WGS submission it is WGS master accession, which in turn links to all contigs.

Submission package have to have all information necessary to create submitted objects without additional communication with NCBI, though it is possible to reference and link objects submitted before.

In case of errors, Submission Portal provides diagnostic message with listing of failed actions and files and provides error descriptions.

# Prerequisites

1. Access to an upload directory on NCBI FTP server or Aspera upload.

Aspera configuration guide is available at: <https://www.ncbi.nlm.nih.gov/books/NBK242625/>

1. A submission group in NCBI Submission Portal. A group should include all individuals who need access to UI-less submissions through the web interface.

Each member of the group should have an individual account. To set-up an account:

1. Log in at [https://submit.ncbi.nlm.nih.gov](https://submit.ncbi.nlm.nih.gov/subs/bioproject/) (Log in button is in the upper right corner)
2. Click on your user name in the upper right corner to update your profile
3. Fill-in your profile and click Save
4. If you registered for direct NCBI account, validate your email
5. Coordinate with NCBI namespace name that will be used in Submitter Provided Unique Identifiers (SPUIDs).

# Submission Package Format

The file submission.xml is an envelope for the whole submission. Its schema is defined in submission.xsd, see <http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/submit/public-docs/common/>.

Here is a brief description of different elements of submission.xml.

### Description element

**Submitter (optional)**

Submitter user name and contact information.

**Organization (required)**

Organization name and contact information.

**Comment (optional)**

Submitter comment.

**Hold (optional)**

If provided, all objects in this submission will not be made public until specified date. This can be overridden for each action.

### List of actions

Each action in the list corresponds to a single submission object. For instance submission.xml could have three actions: one for BioSample, one for BioProject and one for WGS genome.

For the initial submission Action could be represented either by AddFiles element or AddData element.

**AddFiles**

When AddFiles is used, the object data is provided in standalone files that in turn are referenced in AddFiles element. In addition list of key-value attributes is provided to define objects metadata. This method is currently used for WGS and SRA submissions.

**AddData**

Element AddData allows including arbitrary XML block that defines the object. This is being used for BioSample and BioProject which are metadata objects themselves.

Action elements and attributes:

**Status (optional)**

Release policy. Overrides submission Hold policy for the specific action.

**Identifier (required)**

For purposes of UI-less submission SPUID (Submitter Provided User ID - object identifier in submitter’s database) has to be provided. This is used to report back assigned accessions as well as for cross-linking objects within submission. For instance if both BioProject and BioSample are submitted, LocalId can be used to specify the link between them.

When using SPUID spuid\_namespace has to be provided.

**@target\_db (required)**

Target NCBI archive. Currently Submission Portal supports the following UI-less submissions:

1. BioProject
2. BioSample
3. WGS
4. SRA
5. TSA

### Object references

Object reference is used to link different objects within single or multiple submissions. For instance it can be used to link SRA experiment to BioSample and BioProject.

There are three choices for the reference.

**PrimaryId**

The PrimaryId is used when accession of referenced object is known. To specify PrimaryId accession is used as a content of the element and the name of NCBI database as an attribute “db”. For instance:

<AttributeRefId name=”BioSample”><RefId><PrimaryId db="BioSample">SAMN12345</PrimaryId></RefId></AttributeRefId>

List of allowed databases: BioSample, BioProject, SRA

**SPUID (Submitter Provided Unique ID)**

SPUID is used to link objects by their external (user provided ids). SPUID has spuid\_namespace attribute that is unique for each submitter. The values of spuid\_namespace are from controlled vocabulary and need to be coordinated with NCBI prior to submission. Example of SPUID:

<AttributeRefId name=”BioProject><SPUID spuid\_namespace="JGI">JGI12345</SPUID></AttributeRefId>

The id can be an arbitrary string no longer than 128 characters.

# Submission Response Format

Submission Portal response provides updates on submission status for each submission action as well as an aggregate submission status.

The response format is defined by submission-response.xsd, see <http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/submit/public-docs/common/>.

Here is a brief description of response content.

**SubmissionStatus/@status**

Aggregate submission status. Derived from action statuses as described in Appendix A.

**SubmissionStatus/@submission\_id**

An ID assigned by Submission Portal.

**SubmissionStatus/Message**

Error message related to submission.xml envelope.

**Action/@status**

Action status, see Appendix A.

**SubmissionStatus/Action/Response**

Responses for each submission action.

**SubmissionStatus/Action/Response/Message**

Information, warning or error message.

**SubmissionStatus/Action/Response/Object**

Accession and other metadata for submitted object. Submission-response.xsd defines the following attributes for the Object element:

**@accession** – assigned accession

**@url** – link to object in Entrez. There could be a processing delay before object is available in Entrez. HUPed (delayed release) objects will not be available in Entrez until release date.

**@spuid\_namespace** – SPUID namespace that was provided during submission.

**@spuid** – SPUID identifier assigned to corresponding action by submitter.

Content of the Object element is not specified in submission-response.xsd. Check target database samples and documentation for details.

**SubmissionStatus/Action/Response/File**

File attachment(s) produced while processing the action. Could contain error reports, suggested fixes or processing output for instance annotation.

**SubmissionStatus/Action/Response/OriginalFile**

Reference to the file(s) in submission related to this response.

# Submission Protocol

1. Submitter creates a folder (**submission folder**) under his/her **upload directory** for the new submission. The choice of folder name is up to submitter. Submitter can create nested directories to group submissions.
2. Submitter uploads **data files** into submission folder
3. Submitter uploads submission.xml into submission folder
4. After data files and submission.xml are ready to be submitted, submitter uploads submit.ready file into submission folder.
5. NCBI Unified Submission Portal periodically scans upload directory and after it founds new submit.ready (by its timestamp) and submission.xml files it creates a new submission in SP and verifies that all data files referenced in it are present in the submission folder and there is no other unreferenced files (other than submit.ready, report.<N>.xml).
6. If file check (5) is successful SP continues with processing of submission actions specified in submission.xml.
7. Upon completion, SP creates submission report file in the submission.folder. This submission report has name report.<N>.xml, where <N>stands for consecutive numbers 1, 2, etc. SP always start from report file report.1.xml and, if there are more updates on submission state, can create report files report.2.xml, report.3.xml, etc.

Report file contains status for every action in the submission. See complete list of statuses in Appendix A. If some actions have status Processed-error, those need to be corrected and resubmitted. Actions that have status Processed-ok cannot be resubmitted. Once all actions in the submission have status Processed-ok, no further updates to the folder are processed by Submission Portal and this folder could be removed to reclaim space.

SP can delete data files and submission.xml for successful submissions, but leaves all files in submission folder if an error occurred to allow corrections.

No email notification is sent for UI-less submissions.

Submitter has a quota on disk space. If this quota is exceeded, submitter can free some space by removing files from upload directory.

Submitter can also login into SP to see the submission status and stats.

We anticipate having the following type of errors:

1. Not all data files referenced in the submission XML are present. To correct this error submitter need to upload missing files into submission folder.
2. Some data files, which are present in submission folder, are not referenced in submission.xml. To correct this error submitter need either specify these files in submision.xml or delete them from submission folder.
3. Target archive specific errors. To correct these errors, submitter needs to upload new data files and submission.xml.

**Important!** Only failed actions need to be included in update submission.xml. If submission.xml contains actions that succeeded it can lead to creation of duplicate objects.

After all corrections are done, submitter needs to touch (upload) submit.ready file to initiate the processing.

# Appendix A Submission Statuses

### Submission Action Statuses

1. *Queued*

Picked up by target database, automated validations can be rum, but no curator is assigned yet.

1. *Processing*

Transformation, curation and loading.

1. *Processed-ok*

Processing completed successfully, objects are accessioned and loaded in archive. No further re-submissions for this action will be processed.

Accessions are not necessary public yet.

1. *Processed-error*

Processing completed with error(s). Some objects can be accessioned and loaded while some can be waiting for corrections from user.

1. *Deleted*

Action is deleted and no work on it is expected. Could be duplicate, error, etc.

### Submission Statuses

Submission status is determined by combining statuses of actions it contains.

1. If at least one action has Processed-error, submission status is Processed-error
2. Otherwise if at least one action has Processing state, the whole submission is Processing
3. Otherwise, if at least one action has Queued state, the whole submission is Queued
4. Otherwise, if at least one action has Deleted state, the whole submission is Deleted
5. If all actions have Processed-ok, submission status is Processed-ok
6. Otherwise submission status is Submitted