

# Specifications FADA import tool

Version 10/04/2015

Authors: Aaike De Wever, Michel Kapel

This document is used for launching a call for offers for developing the FADA import tool. This document is intended as a guidance for constructing the web application, but as it consists of a complex database and the reconciliation of the needs from 'biologist' users and the technical needs and possibilities, it is likely that specific requirements will have to be clarified and refined along the way.

Supporting material for this specification document can be consulted at  
<https://github.com/aaiakedw/fada-import-specs>.

## 1. General background information

### 1.1 Project status

The Freshwater Animal Diversity Assessment (FADA) database was constructed in 2009 following the publication of a special issue of the scientific journal *Hydrobiologia* in which taxonomic experts described the biodiversity of around 60 organism groups. This work was funded by Belspo and supported by the Belgian Biodiversity Platform.

The FADA initiative and its networking activities involving taxonomic experts led to a follow-up initiative in the form of the EU FP7 project BioFresh (Biodiversity of Freshwater Ecosystems: Status, Trends, Pressures, and Conservation Priorities; [www.freshwaterbiodiversity.eu/](http://www.freshwaterbiodiversity.eu/)), for which RBINS was in charge of constructing a data portal. This project ended officially in April 2014, but we have recently (Spring 2015) finalised the last technical updates (including the implementation of a 'Data Portal Import Tool' – DPIT).

Since May 2014, we focus on the FADA database again through the BRAIN AquaRES (Aquatic Register Exchange and Services) project – [odnature.naturalsciences.be/aquares](http://odnature.naturalsciences.be/aquares). We aim to improve the FADA database, by both streamlining the database import procedures and setting up data exchange with the World Register of Marine Species (WoRMS) and the Register of Antarctic Marine Species (RAMS).

In order to implement this project, we are looking to build a web application “FADA import tool” for importing data into the FADA database.

## 1.2 Database status

Currently (18/11/2014), taxonomic checklists for 16 organism groups are published on-line. In total these checklists contain around 47.000 names and thus represent almost 1/3 of the number of accepted names for freshwater animal species, which is estimated at roughly 150.000 species.

## 1.3 Organisation in groups and volumetry

The organisation by organism group, corresponds to the initial organisation as was adopted for the paper publication. In fact, these groups represent an “operational unit”<sup>1</sup> for which a taxonomic editor was found to produce an informed estimate of the number of known species. These organism groups may represent different taxonomic levels, e.g. classes, orders or families.

For database purposes and/or because no taxonomic editor is found to provide a checklist for the entire group, we may subdivide the task and create a new (sub)group. While taxonomically this represents the creation of a real subgroup, we do not aim to introduce any hierarchical structure at the level of the groups themselves, as these remain “operational units” in the first place.

In terms of size, the checklist for the different groups vary from not even 100 species names to around 16.000 names for Vertebrates-Fish. Unless, at some stage we will further extend the scope at the database to non-animal freshwater groups (other than the “[Macrophytes](#)” group, which is currently included), which is unlikely to happen within the duration of the [AquaRES project](#), the total number of (accepted) species names is unlikely surpass 200.000 and 16.000 for individual groups in the near future.

## 2. Technical background

### 2.1 Technologies used for the BioFresh and FADA database and web applications

The FADA database is managed in PostgreSQL together with the BioFresh occurrence database. The FADA website and the original import scripts were developed in Ruby. The BioFresh portal was developed in the Groovy programming language.

## 2.2 Existing data import tools

For the BioFresh database, we developed an import tool for occurrence data in collaboration with an external developer (Sylvain Renaudier). In essence, the interface for the FADA import tool will be quite similar to that of the 'Data Portal Import Tool' (DPIT). See screenshots 1-3 below.

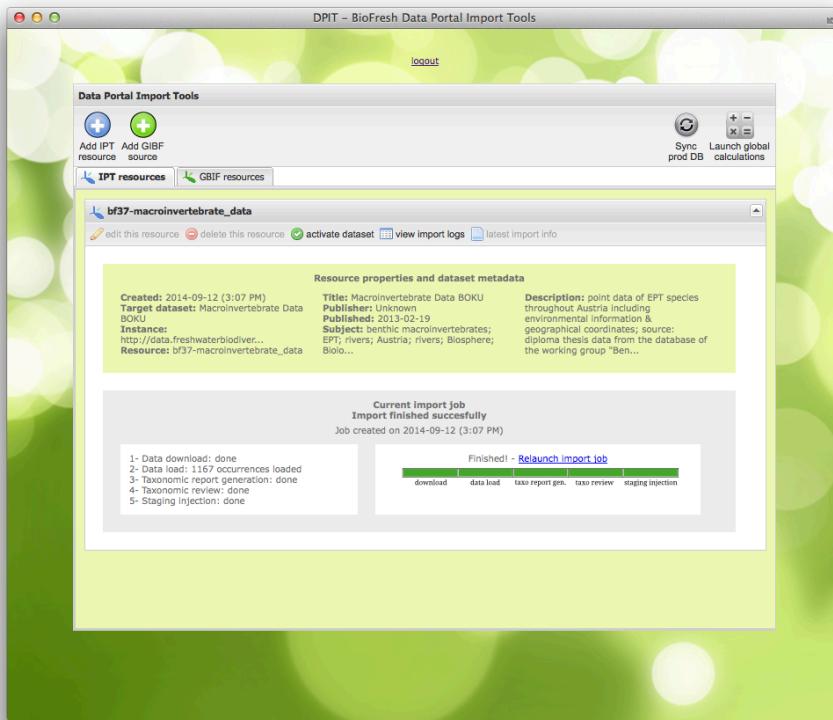


Fig. 1: General user interface of the DPIT.

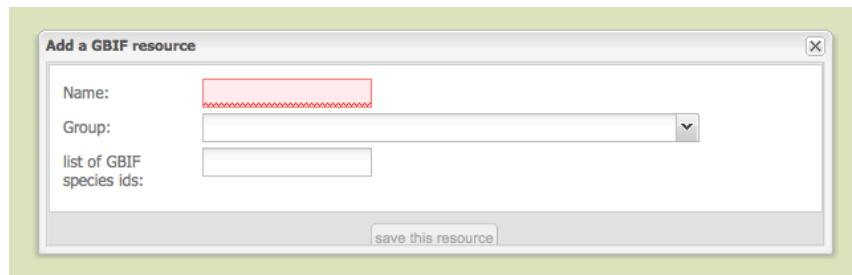


Fig. 2: Resource creation dialog of the DPIT.

The screenshot shows a software application window titled "DPIT – BioFresh Data Portal Import Tools". Inside, a sub-window titled "Dataset report review" is displayed. This sub-window has tabs for "Unknown" and "Known", with "Unknown" selected. It contains a table with the following columns: Action, numberOIO..., gbfDatasetName, gbfDatasetId, matchedBiofreshDatasetName, suggestedBiofresh..., matchedBiofresh..., and largeFile. The table lists numerous rows of data, each representing an import action (e.g., CREATE\_AND...) with its corresponding details like dataset name, ID, and validation status (true/false).

| Action        | numberOIO... | gbfDatasetName                                      | gbfDatasetId                         | matchedBiofreshDatasetName | suggestedBiofresh... | matchedBiofresh... | largeFile |
|---------------|--------------|---|--------------------------------------|----------------------------|----------------------|--------------------|-----------|
| CREATE_AND... | 1742         | United States Geological Survey Nonindigeno...      | d6c0311c-c5ab-4f23-9a20-105149e...   |                            |                      | true               |           |
| CREATE_AND... | 261          | NMNH occurrence DwC-A                               | 5df38344-a821-49c2-8174-e0f294d...   |                            |                      | true               |           |
| CREATE_AND... | 123          | RBINS collections                                   | 8138eb72-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 76           | SINBOTA - marine data (OBIS South America...)       | 83d07ce2-f762-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 25           | Museum of Comparative Zoology, Harvard Uni...       | 4bfac3ea-8763-4fb4-a71a-78af5f5...   |                            |                      | false              |           |
| CREATE_AND... | 23           | Invertebrates (GBIF-SE-SMNH)                        | 56a0a680-0cd0-11dd-84cd-b8a03c5...   |                            |                      | false              |           |
| CREATE_AND... | 20           | UMZC Zoological Specimens                           | 717b3b87-0ebd-46b4-812b-4a2978...    |                            |                      | false              |           |
| CREATE_AND... | 20           | CAS Invertebrate Zoology (IZ)                       | 44bcde48-ac71-4f92-b7f3-24fc3c0...   |                            |                      | false              |           |
| CREATE_AND... | 19           | IndOBIS, Indian Ocean Node of OBIS                  | 8598e6b6-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 17           | Taxonomic Information System for the Belgian c...   | 830b72ac-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 16           | Peabody Invertebrate Zoology DIGIR Service          | 854e35e6-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 15           | Gwali Haanas Invertebrates (OBIS Canada)            | 83c964d4-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 14           | Colección de Cnidarios                              | 84c5b5c0-49f9-41e9-bd0c-c3b85de...   |                            |                      | false              |           |
| CREATE_AND... | 12           | Colección Nacional de Invertebrados - Museo ...     | 8197b330-d9c7-11de-b793-b8a03c5...   |                            |                      | false              |           |
| CREATE_AND... | 12           | Collection Cnidaria SMF                             | 96609212-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 11           | Banco de Datos de la Biodiversidad de la Com...     | 95e635d4-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 11           | Geographically tagged INSDC sequences               | ad43e954-d794-4986-a34-9c0cbfb8...   |                            |                      | false              |           |
| CREATE_AND... | 11           | Zoological Museum Amsterdam, University of ...      | 0dd78500-c066-11dd-a311-b8a03c5...   |                            |                      | false              |           |
| CREATE_AND... | 8            | A Biological Survey of the Waters of Woods Ho...    | 8692f21a-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 7            | SOMBASE BIOCONSTRUCTORS                             | 7b361b5a-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 6            | Sheffield Biological Records Centre - Sheffield ... | da5d29da-804b-42c4-a947-b090d0...    |                            |                      | false              |           |
| CREATE_AND... | 6            | Marine Invertebrate specimen database of Os...      | 720105d5-95ac-4706-bb2b-9f0d420...   |                            |                      | false              |           |
| CREATE_AND... | 6            | Marine invertebrate(ECHINODERMATA) speci...         | 86955830-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 6            | Australian Museum provider for OZCAM                | dce8febb-6c89-11de-8225-b8a03c5...   |                            |                      | false              |           |
| CREATE_AND... | 4            | NODC WOD01 Plankton Database                        | 838b2626-7f62-11e1-a439-00145eb...   |                            |                      | false              |           |
| CREATE_AND... | 4            | naturgucker   | 6ac3f774-d9f8-4796-b3e9-92bfb6c81... |                            |                      | false              |           |
| CREATE_AND... | 3            | Lund Museum of Zoology (MZLU)                       | 427a6290-0c65-11dd-84d2-b8a03c5...   |                            |                      | false              |           |

Fig. 3: Example data validation interface of the DPIT.

For importing data into the FADA database, original Excel-file import scripts were developed in Ruby. With the exception of the scripts for injecting the data into the database tables, the FADA import scripts were re-written in Groovy. Procedures and scripts for importing Darwin Core-Archive (DwC-A) data in FADA have to be developed from scratch.

## 2.3 FADA database structure and planned changes

### SQL scripts current database structure

The definitions of the tables and views in the fada schema are included in the [./database-info](#) folder. For information this folder also contains the definitions for the “importandsyncfada” schema as used for processing Excel imports. Note that this is work in progress, but I included this as it may provide inspiration on how to organise a staging area for processing the import files.

## Description of the main database tables and background with regards to the database structure

The FADA database structure is described at the [BioFresh-wiki](#). The species register (also registry) is described at <http://trac.bebif.be/wiki/BioFresh/DatabaseStructure> under the heading “Organisation of the central species registry”.

The original database structure was elaborated to accommodate a hierarchical browsing tool through the FADA web application (see [FADA Taxonomic Tree browser](#)). Imports were seen as a one-shot thing, where updates would be preceded by a complete deletion of the earlier data, without the need for permanent identifiers. Through BioFresh and exchange projects, the use of the database was widened, and there was a need to introduce more permanent identifiers in order to link the FADA database to the BioFresh species register (hence the *biofresh\_key* tables), and to facilitate the export of the hierarchical data into a flat, denormalised format (hence the *genus\_to\_family* table and the *fst\_for\_genus* and *fst\_for\_tribe* views).

While the organisation of the database may be improved in certain areas, we should maintain the compatibility with the original FADA website [fada.biodiversity.be/](http://fada.biodiversity.be/), as we currently have no plans to rework this application. E.g. while the tables with regards to “observations” are barely used, these correspond to an option in the FADA interface, which we don’t want to break at the moment, and thus should leave those tables untouched.

This means that potential improvements are (in addition to the modifications mentioned below) basically related to the organisation of the *biofresh\_key*-tables and the flat taxon tables. The “biofresh keys” could be integrated in the taxon, species and synonym table eliminating the need for specific tables (this would require a small change in the portal app and export scripts). Instead of having a combination of the the *genus\_to\_family* table and the *fst\_for\_genus* and *fst\_for\_tribe* views, we could opt for extending the *taxons* table with a flat version of the complete taxonomic hierarchy (thus adding the fields: kingdom, class, order, family,... genus, specificEpithet, infraspecificEpithet).

## Planned and needed changes in terms of database structure

In addition to expected changes due to the implementation of a FADA-import tool with its own progress and log database tables, we foresee the need for specific changes based on the required fields to implement the data exchange using the DwC-A exchange format (see 4.1) and certain requests from our editors (categories for aquatic/water dependent species). These changes include the need for the following fields/tables;

- A record level “modified” timestamp, this could be coupled to the import log timestamp
- The isFreshwater (default yes for FADA datasets), isMarine, isTerrestrial, isBrackish fields flags (also to be added to the xls-template) and “Freshwater aquatic/water

dependent” category and subcategory (in xls-template, but currently not stored in the database itself (other than in the import “distr\_table”)). This could be either implemented in the species table or a separate table (cfr. the regions table).

- Need to conserve “provider IDs” for taxons and species names need to be considered. Can this be done by updating the “biofresh key”-tables or should we work out another solution?
- Extend groups table with selected metadata fields from DwC-A EML including; alternateIdentifier for datasetID/URL, keywords and keywordThesaurus, intellectualRights and bibliographicCitation for the dataset as a whole and the URL/location of full EML metadata.

In addition to the required changes described above, we identified a number of issues that merit to be considered to facilitate future developments;

- While the species table currently contains a “status\_id” field and the entries in the synonyms table are by definition all invalid, subjective synonyms, this is not a straightforward solution to provide the “taxonomicStatus” in the DwC-A-export (esp. for providing the “original combination” or “objective synonym”/“basionym” or “homotypic synonym” - terms respectively used in zoology/botany). A solution for this issue should be considered while addressing the FADA database structure. The main question will be whether this issue can be solved without breaking the backward compatibility with the FADA app.
- Bibliographic references are currently organised in 2 tables, *g*references for references pertaining to a group and *publications* linked to individual species (and Rotifera references to be imported are still in a separate schema). Although joining these tables at this stage may not be an option due to the need for backward compatibility, their integration will likely be considered in the future.

## **2.4 “biofresh keys” as persistent identifiers and the link between the FADA database and the BioFresh species register**

As mentioned under 2.3, the “biofresh key” tables originated from the need to have a (more) permanent identifier to establish a link between the FADA database and the BioFresh species register. At this stage there are 3 tables; *biofresh\_key\_species*, *biofresh\_key\_synonyms*, *biofresh\_key\_taxons*, each consisting of an id (considered as “biofresh key”), the id in the respective table for species, synonyms or taxons and the scientific name and/or other crucial data to establish the link with the original tables. Currently, during each update, the ids for the original tables are removed and the new ids are re-established through name-matching. In case of changes in the spelling of the scientific name or changes in the (parentheses of) authorship, this field is updated in the *biofresh\_key* table.

If we design the import tools so that they only update the changed records, we could eliminate the need to get rid of the deletion of the original ids. By integrating the “biofresh keys” in the *species*, *taxons* and *synonyms* table as discussed under 2.3, we would further eliminate the need to update those tables separately.

The link between the BioFresh species register and the FADA schema are covered by the “biofreshdatasources” table (see definition in [./database-info/register-tables](#)) where register.biofreshspeciesregistry.id = register.biofreshdatasources.fkbiofreshid AND register.biofreshdatasources.intdatasourcekey = fada.biofresh\_key[\_species].id. These links remain stable, except if a species name is synonymized and the link to the fada.biofresh\_key\_species.id needs to be replaced by the fada.biofresh\_key\_synonym.id.

For new species names added to the FADA database, their presence in the BioFresh species register needs to be checked (as this register can be populated through other sources) based on the group and species name matching (both exact and phonetic). See further details under 6.2.

### **3. Overall FADA import tool specifications**

#### **3.1 Envisaged components of the FADA import tool**

Data for import into the FADA database will either be available in one of the following formats, which each represent a different *data import module* (corresponding to different “tabs” in the main interface):

1. Data import interface for organism group-data provided in DwC-A format (for externally managed datasets, in particular those coming from WoRMS)
2. Excel template data entry and update interface for FADA checklists (default option for FADA editors)

*Note however that, while data for a group is either present in the database as a “DwC-A” or an “Excel” resource, at some stage we may choose to “throw out” the data imported through an xls-file and import a checklist for a group in DwC-A (if this would be a more complete/trustworthy checklist for example), so there should be a mechanism to delete the data associated with a Excel/DwC-A resource and replace it by a DwC-A/Excel one (could be just creation of a new resource).*

These two modules need to be integrated in a joint interface with the following *shared components*:

1. Resource metadata entry and edit module

2. Functions for synchronising the data imported into the FADA database/schema with the BioFresh species register table
3. Synchronise the staging database with the production database

### 3.2 General user interface

The mock-up displays a user interface titled "FADA import tools". It features two main tabs: "xls resources" and "DwC-A resources". The "DwC-A resources" tab is currently selected, showing a list of categories: Crustacea-Amphipoda, Crustacea-Copepoda, Crustacea-Decapoda-Astacidea, and Crustacea-Isopoda. Each category has a small dropdown arrow icon to its right. At the top right of the interface are two buttons: "Sync BioFresh register" and "Sync prod DB". To the right of the interface, there are two sections of text:

**Note on general user interface:**

- I assume that the interface will be rather similar to the DPIT tool
- while it would be quite useful to have the FADA and data portal import in one tool, I guess this may not be desirable as an older version of ExtJS was used for the DPIT one?
- if possible, I would prefer however that both tools act on the same staging db

**Proposal for resource information shown:**

- info on resources to be stored in the groups table
- add field "harvesting\_type" to distinguish between xls and DwC imports

The "**Sync BioFresh register**" option could be considered as a resource specific function or an overall one. To be decided/discussed.

Fig. 4: The mock-up of the general UI showing the organisation of the two modules in tabs.

- The FADA import tool is by definition a backend tool and requires password protection. A single or fixed user/password (database managers only) should be sufficient in the framework of the current project.
- The user interface should distinguish between the two types of resources: Excel imports and DwC imports (different tab).
- The application will allow the management of data sources (Excel files and DwC-A-files).
- The import process will be independently trigger-able for each resource and associated data source .
- For each resource and associated data source a processing log will be kept, reporting at least the number of lines processed, the number of errors, status information on the processing step indicating successful completion or an error message in case of failure.
- The different import tasks have to be performed sequentially, but are actually independent tasks. The operator can stop between any of these step and resume processing at a later stage.

- During the import phase, the user will be presented with a progress bar to indicate which steps have been completed and which tasks are left.
- The previously developed Data Portal Import Tool (DPIT) uses ExtJS version 3.4, the investment of moving to a more recent version (4.2 or up) should be considered in relation to the potential for integration with and re-use of code from the DPIT on one hand vs. the future cost of maintenance.
- General workflow: Similar to the DPIT, the FADA import tool acts on a staging database, which can easily be copied/synced with the production database. If feasible, we would opt to use the same staging database.

## 4. DwC-A processing module specifications

### 4.1 Envisaged data input format and file location

Input files for Darwin Core-Archive (DwC-A) will obviously be a zipped archive as specified by GBIF (see [here](#)), but will be restricted to files with a “taxon core” and not all Darwin Core fields are intended to be imported in the database. We have selected the required fields for data exchange in the framework AquaRES and have received a sample export from VLIZ. See the [./DwC-file\\_processing/DwC-AquaRES\\_field\\_selection.xlsx](#) file. Selected data fields and extension files will not be mapped to the FADA database, e.g. “nomenclaturalCode” and the “vernacularName” extension. Import scripts have to be constructed. An example export from WoRMS can be provided on request.

*The DwC-A files will be posted on a web address. From our side this may be on an IPT (but these are exports from us and thus do not need to be read by the tool), but I doubt this will be the case for VLIZ. As we are still discussing this, we could of course suggest a way which would make our life easier.*

### 4.2 Main processing steps

#### DwC-A specific processing

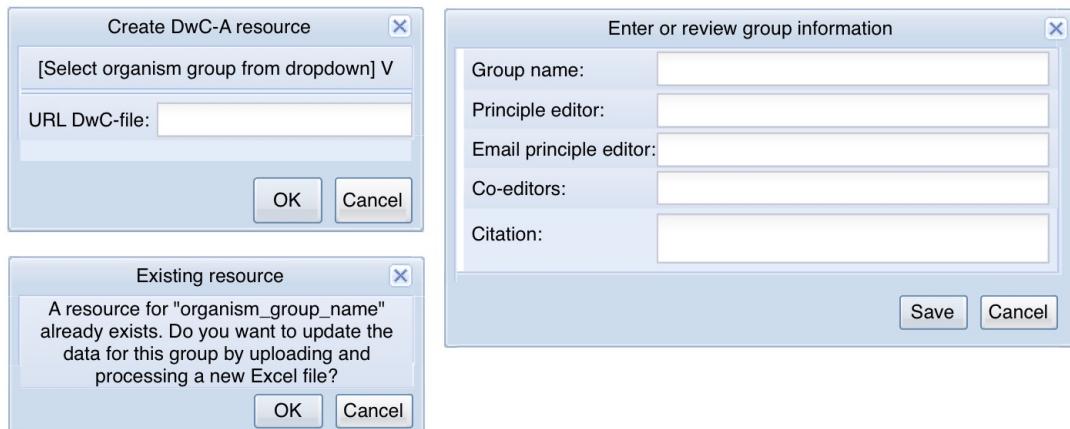
- creating resource and validating metadata (see 4.3)
- data load (see 4.4)
- validation (see 4.5)
- inject data (see 4.6)

#### Processing shared with Excel template processing (see section 6)

- (re)linking to 'biofresh key' tables and re-generation of genus-to-families table (but ideally part of the data injection)
- propagation of changes to BioFresh species register
- synchronisation of staging and production database

### 4.3 Creating resource and validating/editing metadata

Fig. 4 shows the tab for the DwC-import module. Hitting the "Add DwC-A resource" should bring up a dialog to choose the concerned organism group (dropdown) and to indicate whether it concerns a new checklist (entirely new or as a (sub)group of an existing one) or an updated one. See Fig. 5 below.



**Resource creation:**  
- Options in "organism group" dropdown menu:  
-- New group  
-- Group from the groups-table (regardless of whether there are already data available in the database or not)

**Enter or review group information:**  
Auto-populate with info from EML-file and/or groups-table  
If empty, suggests group citation based on URL domain name:  
- For marinespecies.org [to be checked!]:  
WoRMS (Year) "organism\_group\_name". In: Principal editor, Co-editors (Eds) (Year) "Checklist name". Accessed through: http://fada.biodiversity.be/group/show/"groupID" - original data obtained from World Register of Marine Species [date accessed].

File download performed during creation of resource

Fig. 5: mock-up of the resource creation dialog.

At this stage, the operator should also be able to validate that the EML metadata contains the necessary metadata for completing details in the FADA group table (check/enter for FADA specific metadata: editor, co-editor, checklist name,...).

Once finished, the operator should be presented with an overview of the resource and the pending processing steps as in Fig. 6. In this window, we also envisage an “Edit resource metadata” option (along with “View import logs” and “Delete resource”) which allows the editor to edit the resource metadata at any stage. As discussed under 6.1, this metadata editor would be identical for the two modules.

The mock-up shows a web-based application titled "FADA import tools". At the top, there are buttons for "Add FADA xls resource", "Add DwC-A resource", "Sync BioFresh register", and "Sync prod DB". Below these are tabs for "xls resources" and "DwC-A resources", with "DwC-A resources" currently selected. A sub-menu for "Crustacea-Amphipoda" is open, showing options like "Edit resource metadata / View import logs / Delete resource". The main content area displays "Resource properties and checklist metadata" for the Crustacea-Amphipoda checklist, including fields for Checklist name, Citation, Principal editor, Co-editors, Input file downloaded, Number of species, and Distribution level. Below this is a "Current import job" section with a progress bar showing step 2 ("Data load - launch") as completed. At the bottom, there are links to "Cnidaria-Hydriida" and "Crustacea-Cladocera".

#### Processing steps:

- Data download (and metadata loaded) as first step (so this can be relaunched if something went wrong). By default I suggest that this step is automatically performed when creating the resource

#### Proposal wrt. checklist metadata shown:

- Input file update date, validated, published could be integrated in the logs?

Fig. 6: Mock-up showing the overview of the resource processing status.

## 4.4 Data load

Load data in flat taxon, speciesProfile, Reference and Distribution table in staging area.

## 4.5 Validation

### Data in staging tables

*Note: while the validation will act on different tables and fields, and might be slightly different for the DwC-A vs. xls processing, the “business logic” and validation steps are the same, and can thus be organised in a way that the validation steps can be called by the two processing modules.*

- The first “checks” are not really part of the validation process, but are rather “data cleaning” to facilitate further processing. These modifications can be performed automatically and

reported in the logs, but do not need to generate a warning or require intervention of the operator. These include;

- Eliminate empty rows
- Eliminate duplicate rows (keeping only first one)
- Eliminate characters '\n' '\t' '\u00A0' and multiple blanks
- Check for accidental use of "o" instead of "0" and vice versa. E.g. use of "o" in year (199o), use of "0" in latin-only fields (spin0sus). Replace where detected.
- Check whether mandatory fields are present (see overview of DwC-field recommendations as discussed under 4.1) - if error WARNING **mandatory field(s) missing** field\_name, no further processing possible
- Check whether content of fields corresponds to the expected format - if error WARNING **format error** and show line + highlight field, present dropdown menu allowing to ignore/consider empty/edit the field;
  - namePublishedInYear should be of the format 1999a, i.e. 4 digits and optionally one latin character [a-z]. The 4 digits should be >1730 and <current year + 1.
  - The fields kingdom, family, genus, specificEpithet, infraspecificEpithet should only contain latin characters [a-z], accented characters are not permitted
  - The field subgenus should be of the form "Genus (Subgenus)"
  - The field taxonRank should correspond to the control vocabulary: "subspecies", "varietas", "forma", "species", "genus"
  - The field taxonomicStatus should correspond to the control vocabulary: "invalid", "misapplied", "homotypic synonym", "accepted"
  - Check whether the field datasetID contains an URL
  - The fields isMarine, isFreshwater, isTerrestrial, isBrackish and isExtinct, isPreferredName should contain boolean values or values that can be translated to booleans 0/1, t/f, TRUE/FALSE, y/n
  - language should correspond to the ISO 639-1 language code and countryCode should be 2-3 characters (ISO3166 alpha 2 (3 is permissible) country codes)
  - startDayOfYear and endDayOfYear should be between 1 and 365
- Checking for unique taxonIDs in taxon table - if error WARNING **non unique IDs**, show lines/list IDs, offer possibility to ignore a line or abort the import process
- Check whether acceptedNameUsageID and parentNameUsageID refer to taxonIDs present in taxon table - if error WARNING **core IDs missing from taxon table**, show lines/list IDs, offer possibility to ignore a line or abort the import process
- Checking whether taxonIDs/coreIDs in extension tables correspond to taxonIDs in taxon table - if error: as above
- Checking the consistency of the row, e.g.: if the "specificEpithet" field is provided, the "genus" field cannot be empty. If error WARNING **line consistency problem for 'taxonRank' missing 'higherTaxonRank'**: show line and offer possibility to ignore line or edit line

- Check whether data for higher taxonomic levels are correctly declared; genus requires family, species requires genus, subspecies requires specificEpithet. - if error WARNING **higher taxonomy missing in input file for** taxonRank:scientificName missing taxonRank:scientificName, offer possibility to ignore or edit line. See Figs. 7 for an example mock-up.

**Validation report review**

Field errors Declaration errors Import: new Import: update Import: conflict

| Action      | Error type   | Description         | Field   | Field_content     |
|-------------|--------------|---------------------|---------|-------------------|
| IGNORE LINE | format error | no valid year       | Year    | & Lambrechts 1987 |
| IGNORE LINE | format error | non-latin character | Species | capernaumi'       |
| EMPTY       |              |                     |         |                   |
| EDIT        |              |                     |         |                   |

**Example for within line and within file consistency for data in the Excel file template (but same is the case for DwC-A files):**

• For the species *Halacarellus fontinalis* to be correctly declared, (a) the 3rd line needs to contain the information on the higher taxon levels and (b) any higher taxon levels which are documented on the 3rd line (in this case for Genus and Family) need to be declared on a separate line.

| Family      | ... | Genus        | Subgenus | Species group | Species    | Subspecies | Author(s)        | Year  |
|-------------|-----|--------------|----------|---------------|------------|------------|------------------|-------|
| Halacaridae |     |              |          |               |            |            | Murray           | 1877  |
| Halacaridae |     | Halacarellus |          |               |            |            | Viets            | 1927b |
| Halacaridae |     | Halacarellus |          |               | fontinalis |            | Bartsch & Ger... | 2011  |

**Validation report review**

Field errors Declaration errors Import: new Import: update Import: conflict

| Action  | Error type               | Description                   | Family      | ... | Genus        | ... | Species    |
|---|--------------------------|-------------------------------|-------------|-----|--------------|-----|------------|
| EDIT  | line consistency         | genus field empty             | Halacaridae | ... | Halac        | ... | fontinalis |
| IGNORE LINE                                     | hierarchical consistency | parent for genus not declared | Halacaridae | ... | Halacarellus | ... |            |
| <input checked="" type="checkbox"/> CREATE FROM |                          |                               |             |     |              |     |            |
| <input checked="" type="checkbox"/> CREATE NEW  |                          |                               |             |     |              |     |            |

**Validation report review**

Field errors Declaration errors Import: new Import: update Import: conflict

| Database   |                 |     |                 | Import |                 |     |                  |      |
|--|-----------------|-----|-----------------|--------|-----------------|-----|------------------|------|
| Action   | specificEpithet | ... | Author(s)       | Year   | specificEpithet | ... | Author(s)        | Year |
| APPLY UPDATE                                     | fontinales      | ... | Bartsch & Ger.. | 2011   | fontinalis      | ... | Bartsch & Ger... | 2011 |
| IGN. UPDATE                                      | inopinatus      | ... | Fain            | 1987   | inopinatus      | ... | Fain & Lambr...  | 1987 |
| <input checked="" type="checkbox"/> APPLY UPDAT  |                 |     |                 |        |                 |     |                  |      |
| <input checked="" type="checkbox"/> EDIT & APPLY |                 |     |                 |        |                 |     |                  |      |

Fig. 7: Mock-up of the validation interface

#### Imported data compared to data in the database tables

Based on the groupID (entirely new group?) and (provider) taxonID/coreID > Check which data are already present in the database and compare content of fields if the provider taxonID/coreID is already present. Records can either be;

**NEW:** Alert the operator that this is new (unless it is an entirely new group).

**UPDATED:** Associated information added (e.g. distribution and speciesProfile data previously not available). Alert operator, default option “Apply update”, option for operator to “ignore update” .

#### POTENTIAL CONFLICT:

This type of error mainly applies for any name related fields including taxonID, scientificName, genus, specificEpithet, etc. covered by the next checks.

In case of errors, the operator should be presented with the options to ignore (keep former data), override (adopt new data) or edit and override (adopting newly entered data).

- Check whether records with the same provider taxonID (and resourceID) in the species and synonyms table have the same “scientificName, acceptedNameUsageID, parentNameUsageID, acceptedNameUsage, originalNameUsage, parentNameUsage, namePublishedIn, namePublishedInYear, kingdom, phylum, class, order, family, genus, subgenus, specificEpithet, infraspecificEpithet, taxonRank, scientificNameAuthorship”
- If the taxonID is new, check whether the “scientificName” can be found in the species or synonyms table and whether the status is “accepted” for names in the species found in the species table and “invalid” for names found in the synonyms table.
- If the scientificName is not detected, check whether the combination/concatenated names of the “genus” “specificEpithet” and “infraspecificEpithet” can be found in the species or synonyms table and whether the status is “accepted” for names in the species found in the species table and “invalid” for names found in the synonyms table.
- If the taxon/scientificName is not detected using exact matching during the 2 previous steps, repeat with phonetic matching.
- Check whether scientificNames with taxonRank =< family are new or are updated (for a new resource this should be the case, but during an update, the operator would be interested to know what's new and what was already present in the database).
- Check whether scientificNames with taxonRank > family need to be added or are already present in the taxon table (check for conflicts) - if error WARNING **conflicting higher taxonomy**, show conflicting levels + names, provide options: ignore (keep original higher taxonomy), override (adopt new higher taxonomy) and edit and override (edit and adopt new higher taxonomy). Currently the higher taxonomy is copied from MySQL database tables on the [annual release of the Catalogue of Life](#) DVD. A more sustainable solution for this issue, using the Catalogue of Life web services should be investigated.

See lower panel of Fig. 7 for a mock-up example of the validation interface for conflicting changes.

## 4.6 Data injection

Inject data in database tables and update biofresh\_key tables. Updating the biofresh\_key table requires to build an overview of the updated names (e.g spelling corrections), which need to keep their original biofresh\_key, the new species, for which a new id is generated, and the deleted ones, for which the logical delete flag needs to be set. See more background details on the biofresh\_key tables under 2.4.

## 5. Excel template processing

While scripts for importing data are currently available for importing Excel data, it would probably be more efficient to re-write them. Nevertheless, these scripts could provide inspiration for how to tackle specific issues. More details can be found in the “Current workflow: Excel template processing” section of an earlier version of the specifications document [FADA-import-specs.md](#).

**FADA import tools**

Add FADA xls resource   Add DwC-A resource

**xls resources**   **DwC-A resources**

Sync BioFresh register   Sync prod DB

Arachnida-Halacaridae  
Cnidaria-Hydrida  
Crustacea-Cladocera  
Crustacea-Copepoda  
Crustacea-Mysidacea  
Crustacea-Ostracoda  
Insecta-Ephemeroptera  
Insecta-Plecoptera  
Insecta-Trichoptera  
Macrophytes  
Mollusca-Bivalvia  
Nematomorpha  
Platyhelminthes-Turbellaria  
Rotifera  
Vertebrates-Fish  
Vertebrates-Mammals

**Note on general user interface:**

- I assume that the interface will be rather similar to the DPIT tool
- while it would be quite useful to have the FADA and data portal import in one tool, I guess this may not be desirable as an older version of ExtJS was used for the DPIT one?
- if possible, I would prefer however that both tools act on the same staging db

**Proposal for resource information shown:**

- info on resources to be stored in the groups table
- also show groups that are present in the db, but have not previously been created/processed through the web tool
- add field "harvesting\_type" to distinguish between xls and DwC imports
- only show groups which have been published

The "**Sync BioFresh register**" option could be considered as a resource specific function or an overall one. To be decided/discussed.

Fig. 8: Mock-up of general UI for Excel resource processing.

### 5.1 Envisaged Excel input format

Excel-data are provided in an [excel-template](#), for which there are currently 2 “recognised” versions. A slightly modified version (2.1) including the columns isFreshwater, isMarine, isTerrestrial and isBrackish will need to be prepared reflecting the planned database changes (see 2.3). While editors previously had the possibility to provide checklist data through multiple files for a single group/resource, we may choose to abort this option for simplicities sake (if necessary we can still manually join the files).

## 5.2 Main processing steps

### **xls specific processing**

- creating resource and validating metadata (see 5.3)
- checking column mapping (see 5.4)
- data load (see 5.5)
- validation (see 5.6)
- inject data (see 5.7)

### **Processing shared with DwC-A processing (see section 6)**

- (re)linking to ‘biofresh key’ tables and re-generation of genus-to-families table (but ideally part of the data injection)
- propagation of changes to BioFresh species register
- synchronisation of staging and production database

## 5.3 Creating resource and entering metadata

*See the UI-mock-ups in Fig. 9 and 10.*

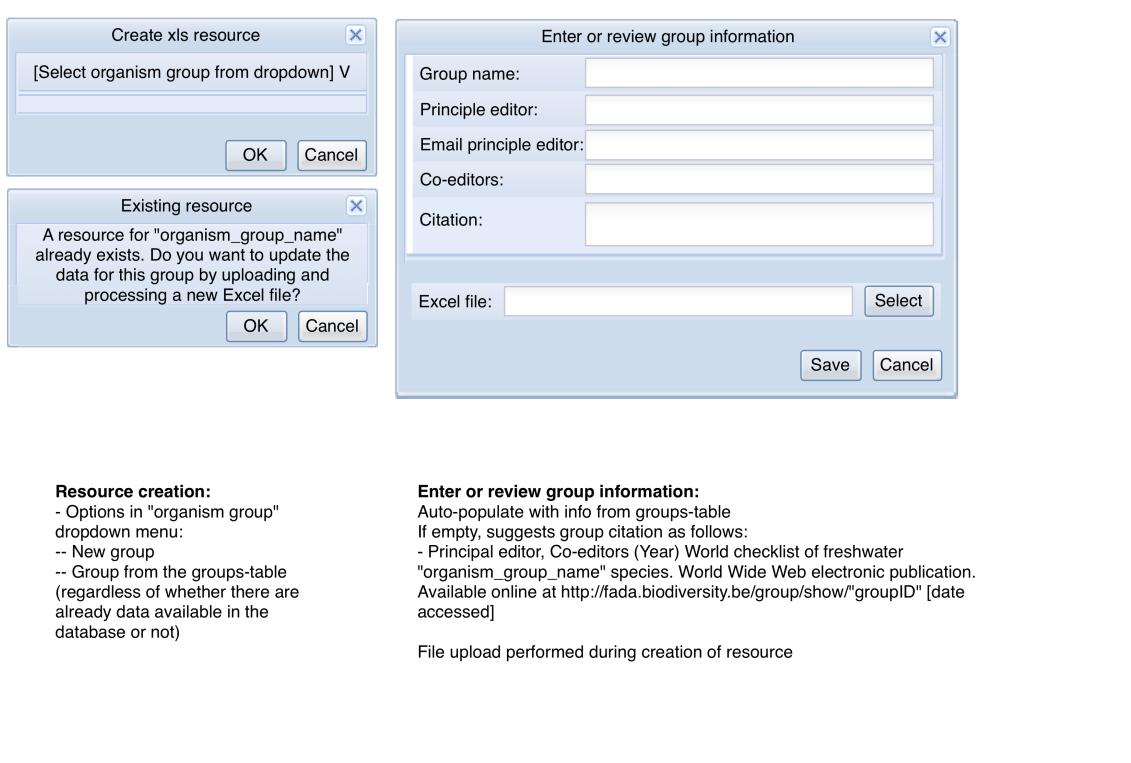


Fig. 9: Mock-up of the Excel resource creation dialog.

The interface is titled "FADA import tools" and includes the following components:

- Top navigation:** Add FADA xls resource, Add DwC-A resource, Sync BioFresh register, Sync prod DB.
- Tab navigation:** xls resources (selected), DwC-A resources.
- Resource list:** Arachnida-Halacaridae (selected). Options: Edit resource metadata / View import logs / Delete resource.
- Resource properties and checklist metadata:**
  - Checklist name: Arachnida-Halacaridae
  - Created: dd/mm/yy
  - Updated: 21/01/13
  - Agreed for publication: yes
  - Citation: Bartsh, I., 2013 Jan 21, World checklist of freshwater Halacaridae species. World Wide Web electronic publication. Available online at <http://fada.biodiversity.be/group/show/32> [date accessed]
  - Principal editor: Bartsh, Ilse <bartsch@meeresforschung.de>
  - Co-editors: /
  - Input file received: dd/mm/yy
  - Number of species: 56
  - Distribution level: /
- Current import job:**
  1. Check column mapping - [launch](#)
  2. Data load
  3. Validation report generation
  4. Validation report review
  5. Staging injection
- Bottom navigation:** Cnidaria-Hydroidea, Crustacea-Cladocera.
- Right sidebar:** Proposal wrt. checklist metadata shown:
  - Input file update date, validated, published could be integrated in the logs?

Fig. 10: Mock-up of resource status overview after creation.

During the creation of an Excel resource, the operator can create a resource for an organism group present in the groups table (cfr. [http://fada.biodiversity.be/group/list?current\\_page=groups](http://fada.biodiversity.be/group/list?current_page=groups)) or create a new (sub)group as detailed for DwC-A resources under 4.3.

In case of an existing group for which information is currently available, the operator will be presented with a warning to highlight the fact that this resource already exists. Further processing will not create a new resource, but continue with the review of metadata and upload of an Excel file using the existing info on the resource (also allowing it to be updated).

For a group present in the list, metadata present in the group-table will be included in the dialog and can be reviewed and completed at this stage. As the Excel-file does not contain checklist metadata, these fields will be empty and need to be filled for a new resource. With the exception of the field “co-editors”, this information is mandatory.

We propose to include file upload (from the operator's computer) as part of resource creation. In case of file upload errors, the application should offer a retry option before closing the resource creation window and process.

## **5.4 Checking column mapping**

The use of Excel files makes it impossible to be certain of the structure of the files that are sent to us. Contributors can make errors in data structure, field positions, start of data in excel sheets, etc. It is therefore necessary for the operator to make sure that the files comply with one of the two templates that have been agreed on. Essentially this is a check of columns and data position. The easiest solution seems a quick visual checking mechanism to validate the field mapping. This is illustrated in Fig. 11 and 12.

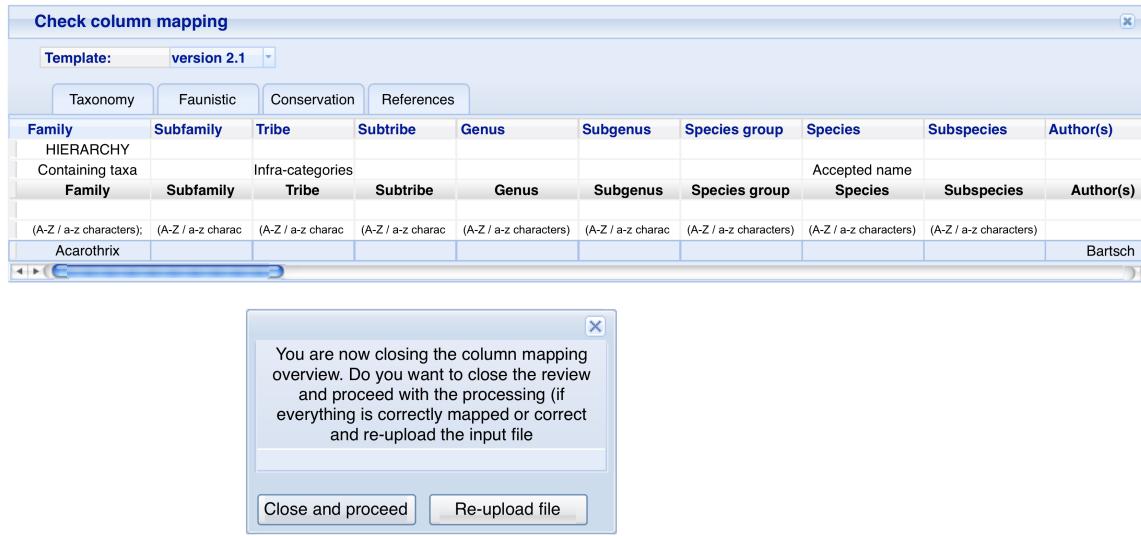


Fig. 11: Mock-up column mapping interface.

The image shows the 'FADA import tools' interface. At the top, there are buttons for 'Add FADA xls resource', 'Add DwC-A resource', and sync buttons for 'BioFresh register' and 'Sync prod DB'. Below this, tabs for 'xls resources' and 'DwC-A resources' are visible, with 'xls resources' currently selected. A checklist for 'Arachnida-Halacaridae' is displayed, showing metadata like 'Checklist name: Arachnida-Halacaridae', 'Created: dd/mm/yy', 'Updated: 21/01/13', and 'Agreed for publication: yes'. It also lists citation details, principal editor (Bartsch, Ilse), co-editors, input file received, number of species (56), and distribution level. A 'Current import job' section lists steps 1-5: 'Check column mapping - open', 're-upload file', 'Data load', 'Validation report generation', 'Validation report review', and 'Staging injection'. At the bottom, other checklist entries are listed: 'Cnidaria-Hydrozoa' and 'Crustacea-Cladocera'.

Fig. 12: Mock-up of the resource status overview after completion of the column mapping.

## 5.5 Data upload

The upload step is the reading data from Excel files and storing it in tables in a staging area.

Upload happens per group. The files to process are all located in a directory specific for the group/resource (named after the group's name) and all loaded one by one.

The upload process looks for three sheets of data are identified based on their names, which can already be validated while checking the column mapping. Each sheet – “Taxonomy”, “Faunistic” and “References” respectively – is stored in a dedicated import table (taxa\_table, distr\_table, ref\_table).

## 5.6 Validation

### Data in staging tables

- For data entered in the Excel template, we do not use the notion of “mandatory fields”. Nevertheless, it would be useful to flag whether author and year are empty on specific lines. This could be flagged as **recommended data missing**: ‘year’/‘author’ not provided for line xx.
- While the data are rather different in terms of organisation, validation for **format errors** the corresponding fields is the same as 4.5.
- If present in the “References” the presence of the Refkey in the “Taxonomy” sheet has to be checked.
- As data in Excel format does not come with “Core IDs”, the appearance of the names from the “Faunistic” sheet has to be checked.
- As in 4.5, there’s a need to check the **line consistency** and whether data for **higher taxonomy is missing**.
  - Particular for the Excel templates is that synonyms are not declared in the same format as the accepted species, but are present in dedicated columns. To be validly declared, the accepted species to which the synonym refers has to be declared on a separate line (similar to higher taxonomy levels). However, as this is easily overlooked by the editors, I propose to implement the option to **declare species based on information on line xx** as default option.
  - Another line consistency check particular for data in the Excel-format is checking whether the “original\_genus” is provided when the parentheses flag is set to “yes” and/or if “declension\_species” is given.

### **Imported data compared to data in the database tables**

The validation in comparison to data already in the database is also similar to 4.5, with the exception that it cannot be done based on IDs, and can exclusively be performed through name matching.

## **5.7 Import**

As 4.6.

# **6. Joint components**

## **6.1 Resource metadata entry and edit module**

Similar to metadata inspector/editor under 4.3 but including all fields from the dataset table (e.g. abstract/description) and the new ones described under 2.3.

## **6.2 Synchronise data with BioFresh species register**

As a “final calculation” function, there is a need for synchronising the data imported into the FADA database/schema with the BioFresh species register table. In addition to the requirements needed for updating the biofresh\_key tables, this also requires an overview of the updated “original combinations”. In contrast to the species and synonyms, which are stored in a specific table, there is no separate table for these “original combinations”. A list has to be constructed by combining the information from the following fields; original\_genus, declension\_species or species (if the former is empty), year and author.

As mentioned under 2.4, for new species names added to the FADA database, their presence in the BioFresh species register needs to be checked (as this register can be populated through other sources) based on the group and species name matching (both exact and phonetic).

This will likely require an interface for validating the changes; (a) accept/ignore/link to existing for new entries, (b) check exact matches and accept to update “name source” to FADA, and (c) check phonetic matches and update name from FADA/add FADA name and keep name from register as synonym.

### **6.3 Synchronise the staging database with the production database**

In parallel to the developments of the FADA import tools, we will work out a solution to improve the synchronisation/replication of the staging database (for both the DPIT and FADA import tool) with the production database. If successful, we will provide the code to trigger the synchronisation for use with the “sync prod DB” button in the general interface.

---

1. The “operational unit” can be referred to as “Group”, “Data source” or “Resource”. In this document the term “group” is used to stress the link with taxonomic unit, whereas “resource” is used when discussing the digital representation of such a group (e.g. as an ‘import unit’ and its associated file). [←](#)