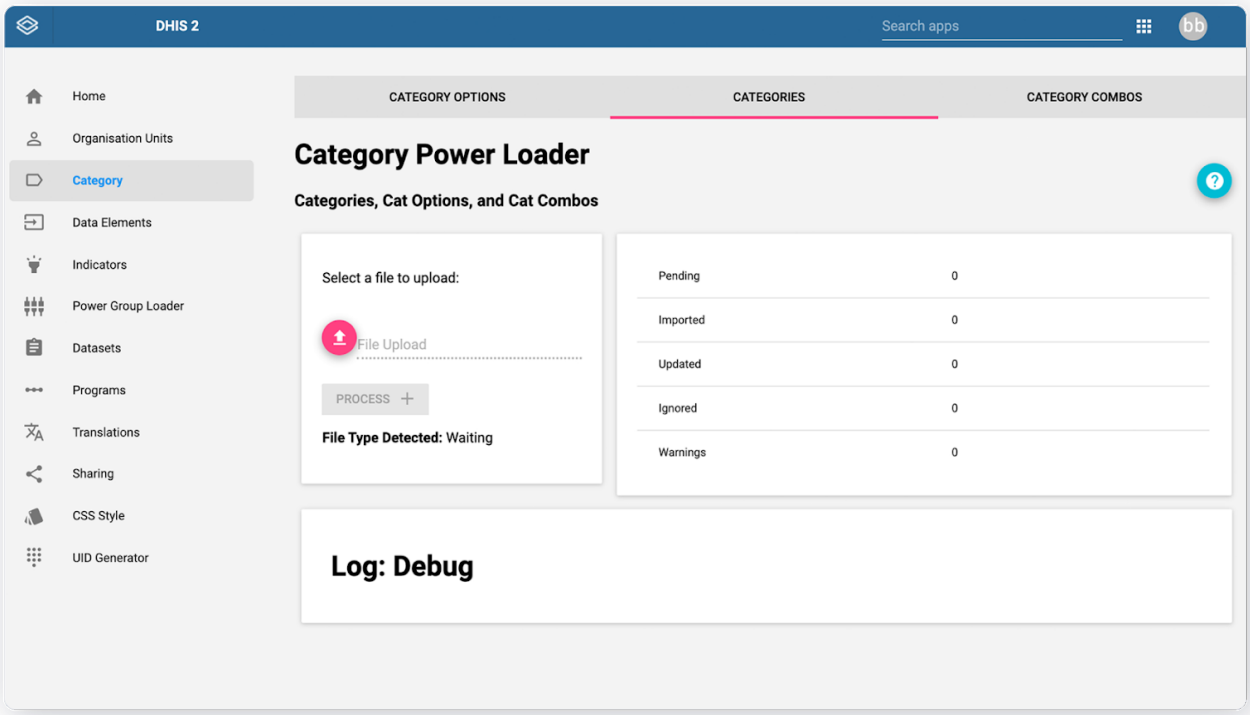
bao systems

BAO Import Foundry (BIF)

User manual



2900 K St NW, Suite 507
Washington, DC 20007
+1 (202) 536-1543

info@baosystems.com
www.baosystems.com

Table of contents

What is the BAO Import Foundry?

The BAO Import Foundry (BIF) is a user-friendly configuration tool used to quickly and easily create metadata objects in DHIS2 in bulk upload form. Using a standard format csv file uploaded via BIF by the user, the metadata objects are then created in DHIS2, saving you time bypassing complex API configuration, or directly using the DHIS2 interface.

BIF is made up of a series of modules, each designed to create specific metadata objects in DHIS2 form. The tool allows you to create all of your metadata objects, which may be in the hundreds, in one go.

Currently, BIF contains the following modules:

- Organisation Units
- Category Power Loader
- Data Element Loader
- Indicator Loader
- Power Group Loader
- Dataset Loader
- Program Loader
- Translations
- Sharing
- Style Loader
- UID Generator

To see future models, go to the Roadmap section.

It is important to note that BIF should not be used to update objects, with the exception of data elements. Objects with high dependencies like programs, data sets, and groups can be affected by such updates. For example, if a data element group already exists, attempting to update that group via BIF will overwrite all existing data elements within that group, unless they are also specified in the upload file. Further, some objects like program stage data elements are system generated and not specified in a BIF update template. Updating the program stage elements using BIF may result in the creation of new objects and render some orphaned. Yet another danger is that any pre-existing sharing settings on an object will be erased if an update is made using BIF.

There is a partner app to BIF, called dAtaZ, which is capable of handling the bulk import of aggregate, events and tracker data in DHIS2, through a standardised flat CSV import template. Please reach out to a BAO representative if you are interested in learning more about dAtaZ.

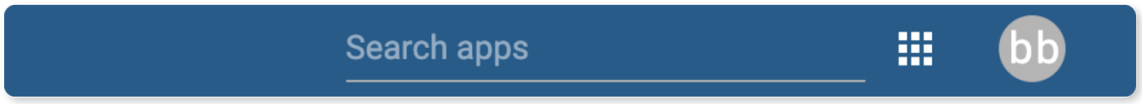
Getting started with BIF

In getting started with the BIF tool, we must first install the application as it does not come pre-installed with DHIS2. To install the BIF, the system administrator must have access to the BIF install Zip file.

Installing BIF

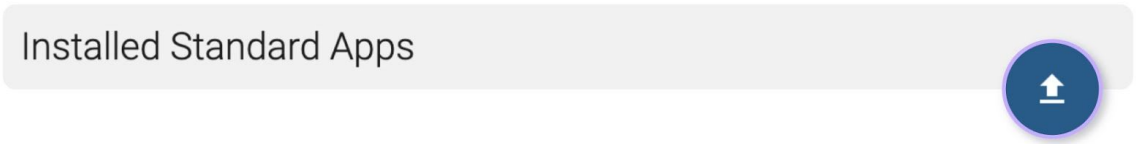
BIF is installed through the App Management module of DHIS2. To install:

1. Hover over (or click) the **Apps** button in the upper right corner on the header bar. Type in “App” in the “Search apps” search bar.



2. Click the **App Management** icon.

3. In the Installed Standard Apps section, click the **Upload** icon.

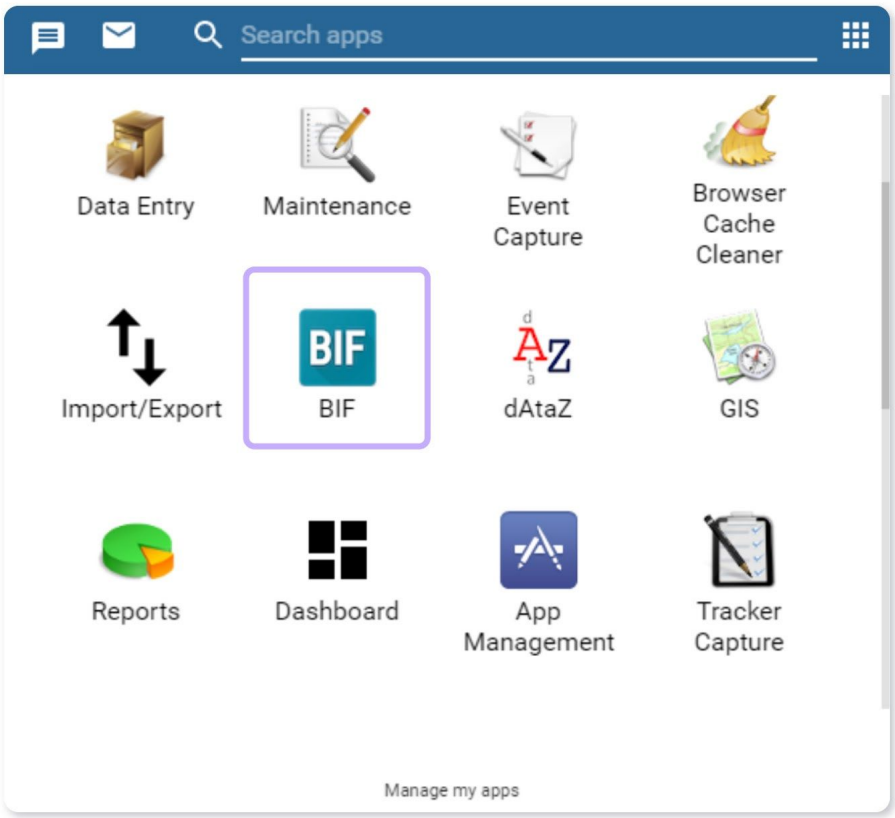


4. After clicking upload, navigate to the folder where the BIF install zip file is on your local system and select it.
5. Once successfully installed, restart or refresh your page and BIF should be listed among the installed apps as shown below.

Accessing BIF

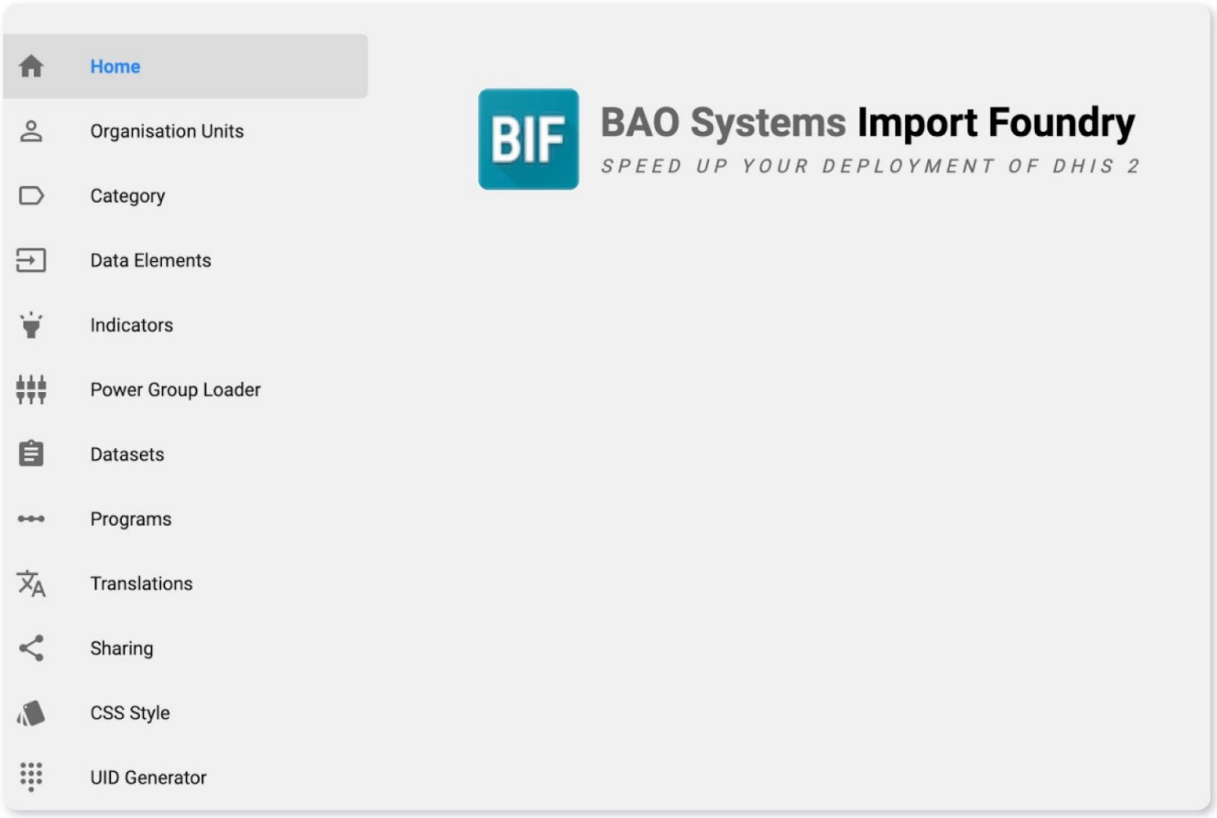
BIF is accessed like every other built-in DHIS2 app. To access BIF:

1. Hover over (or click) the **Apps** button in the upper right corner on the header bar.
2. Scroll down and select BIF from the menu.



Modules within BIF

You now have access to the numerous modules within the application:



- **Organisation Units** - This module is used to create Organisation Units.
- **Category (Loader)** - This module has 3 tabs which allows for the upload of Category Options, Categories, and Categories Combinations respectively.
- **Data Elements** - This module allows the uploads of data elements.
- **Indicators** - This module allows for the upload of indicator types.
- **Power Group Loader** - This module uploads different types of group objects in DHIS2, depending on what the user specifies as the group type for uploading in the CSV file used for the import. It can be used to upload organisation unit groups, data element groups, category option groups and indicator groups. If you are updating an existing group you must specify the entire group members or existing members will be overwritten and removed at import.

- **Datasets** - The datasets module allows the upload of datasets and sections.
- **Programs** - This module has 3 tabs for the upload of Event programs, Tracker programs and Program Stage Sections respectively.
- **Translations** – This module allows the user to add, delete, replace or update translations.
- **Sharing** - This module allows objects to be shared within DHIS2. It creates and updates the external access, public access, and specific user groups with which the user can share within an object.
- **CSS** - Allows for the import of CSS stylesheets when customising DHIS2 appearance and the login page.
- **UID Generator** - UIDs are important in the configuration of metadata in DHIS2; every object in the system has an associated unique ID. This module enables the generation of DHIS2 accepted UIDs. To help keep track of the objects when they are being created, as they may need updating later on, it is advisable to include all UIDs in the CSV file for import during configuration.

Using the BIF modules

Principles

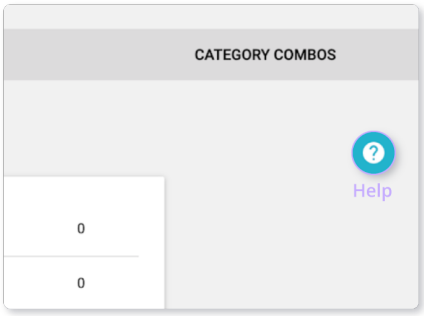
Every single module of the BIF tool operates in the same way:

- The accepted import format is a CSV template. We recommend a UTF-8 CSV to preserve special characters.
- The arrangement of each template is outlined in the help text, as are any possible range of options for how to complete it.
- It is possible to copy the template headers from the bottom of each help text into a spreadsheet and convert the text into columns, after which you will have a blank template ready to populate.
- The properties in a template reflect the possible properties available in the GUI of DHIS2 for a particular object.
- Each module within BIF is laid out in the same way, with a few minor differences depending upon the object. It is helpful to establish an understanding of the different options outlined in the app:

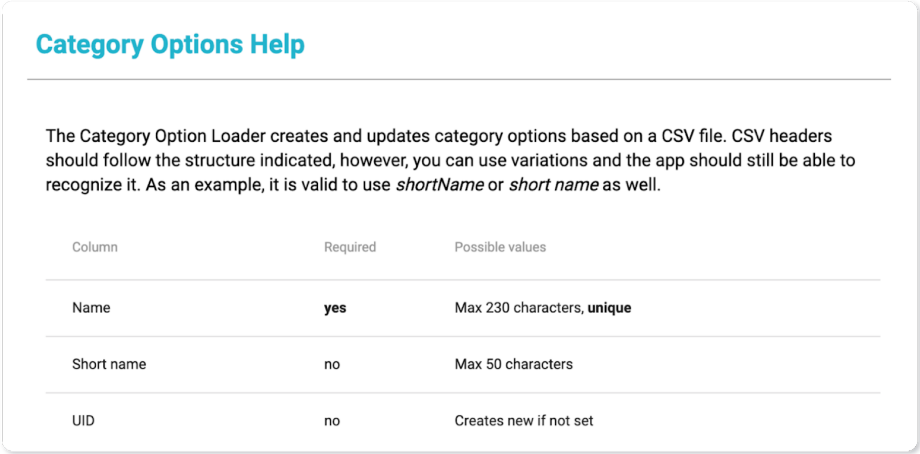
- **Upload layout** - using this function the user can select files for import, customise settings for your object type and also, in some cases, carry out a dry run import by use of the toggle.
- **Help text** - this feature reveals the help text for template population.
- **Result layout** - once a file is selected for import, a validation will read the file and show an expected response. Clicking on any area of the response will expand the logs. Post-import, this area also shows the final outcome of the import.
- **Log layout** - this displays the logs for any activity clicked upon in the result layout. This is particularly helpful if there are ignored or warning responses.Which when selected will be outlined for the user in the log area.

The help text

The “help tool” provides guidance on accepted CSV template structures to use when importing templates for each module. It is a guide for the successful metadata upload for users to reference when populating their BIF template. Represented by the “?” icon located at the top right of the page, this tool is available within each module of the BIF app.



Within the “Category” module of the BIF app, when you click on the help tool the following box pops up.



Category Options Help

The Category Option Loader creates and updates category options based on a CSV file. CSV headers should follow the structure indicated, however, you can use variations and the app should still be able to recognize it. As an example, it is valid to use *shortName* or *short name* as well.

Column	Required	Possible values
Name	**yes**	Max 230 characters, **unique**
Short name	no	Max 50 characters
UID	no	Creates new if not set

The help text is split into three sections:

- A list of all the properties possible to draw upon when populating a template. In the help text, each row is a different property and the name of the property in the first column of this list should correspond to the header used in the BIF template. As many properties are optional, the list also tells the user if a property is required in the BIF template, and any possible/permissible values that can be used when completing the template for a given property.
- An example of a template, which provides an example of how to complete a template. It provides guidance as to what a final completed template may look like.
- CSV headers. This is a list of all the headers, in their correct format for import, for every property within the object template. This is useful for the user to copy into a spreadsheet. Since it is a comma separated list, you can split the text into columns and as a result create a pre-formed blank template. Then you can decide to remove the columns that are not needed and populate the ones that are."

Helpful hints

Not every property used to populate the template needs to be present in the final import version. For example, you may want to upload data elements with no codes. In such cases, where a property is not required and you are not using it, you can simply exclude the property from your template.

It is good practise to keep the order of the columns the same as is specified in the help text. While not strictly necessary, it is the most reliable method for import and for keeping track of the data in general.

It is required that the headers for your template, those values in the first row representing properties of the object you wish to create, match precisely with the format shown in the BIF help text. Thus, if you have a header in the help text which says “shortName” it is not acceptable to use “short name”, “short name” or “Short Name”. Such attempts will return an error. It is easiest to simply copy the CSV list at the end of the help text and complete the sheet using the format outlined.

The BIF tool requires a template for metadata upload; this is called the BIF template.

The BIF template is used to define the values for the properties required for a successful bulk metadata import.

The BIF template has headers that mirror possible properties for the objects being created. For example, the image below shows the Data Element properties from the GUI and in the BIF template.

BIF template header

Name	Short name	Domain type	Value type	Aggregation type	UID	Code	Description	Category combo	Form name	Zero is significant	Option set

Object properties from GUI

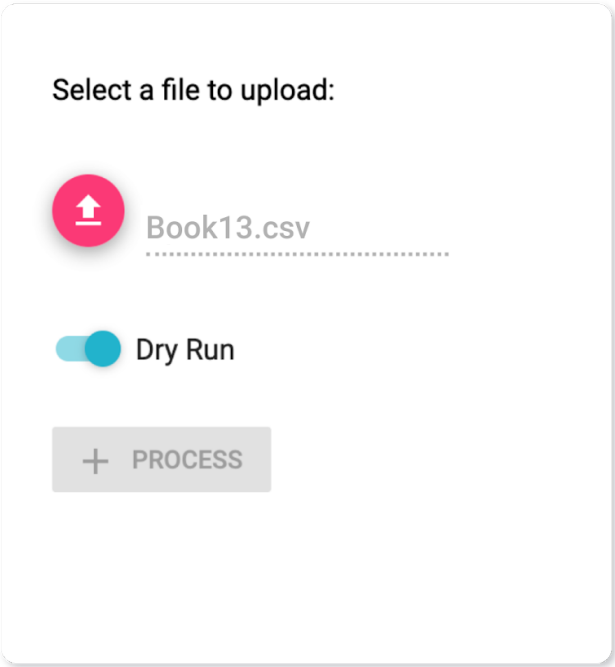
Name	Short name	Code	Description	Form name	Domain type	Value type	Aggregate type	store zero values	Category combination	Option set

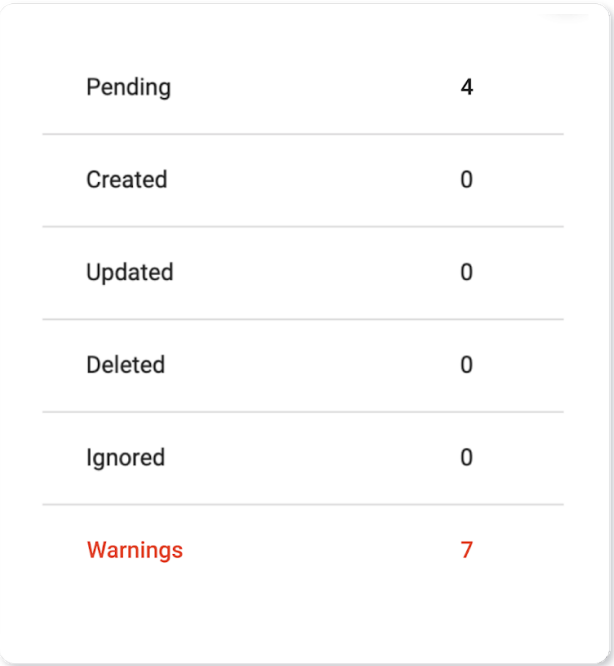
Here is an example of a populated data element BIF template:

Name	Short name	Domain type	Value type	Aggregation type	UID	Code	Description	Category combo	Form name	Zero is significa	Option set	Attribute:jO7FtVvBEXL
XY-SN4 System type	SN4: System type	TRACKER	TEXT	NONE	Gx7FkFESQIF				System type	FALSE	G7FbaccY9Pk	
XY-SN4 Policy name	SN4: Policy name	TRACKER	TEXT	NONE	TG5rQ0vtHXW				Policy name	FALSE		tQiqZmeRuCf
XY-SN4 Policy type	SN4: Policy type	TRACKER	TEXT	NONE	BEqxI8ppPIf				Policy type	FALSE	Qc7vPFzEAxm	Wu3vIgbwg99
XY-SN4 Policy sector	SN4: Policy sector	TRACKER	TEXT	NONE	c7GAVBigpX9				Policy sector	FALSE	sVMEcxgZb8I	PPpqbJAGdyt
XY-SN4 Positive influence	SN4: Positive influence	TRACKER	BOOLEAN	COUNT	Lm0uaPKNjTi				Positive influence?	TRUE		BrLVjUdZxv3
XY-TGT-SN4 Target number	TGT-SN4: Conflict/fragility EV	AGGREGATE	INTEGER_ZERO	SUM	ThOWd3GpxUe			ySy2zZakOEc	TGT-SN4: Number c	TRUE		

The import area

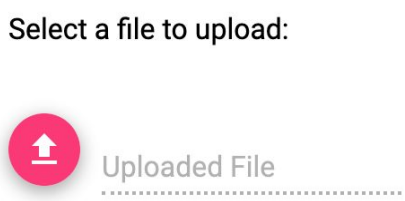
The import area of the BIF tool is where the BIF template is uploaded and the result of the upload is displayed. The result provides an analysis of the upload, before and after the upload.





Steps to upload metadata:

1. Click on the red/pink button to upload the BIF template.



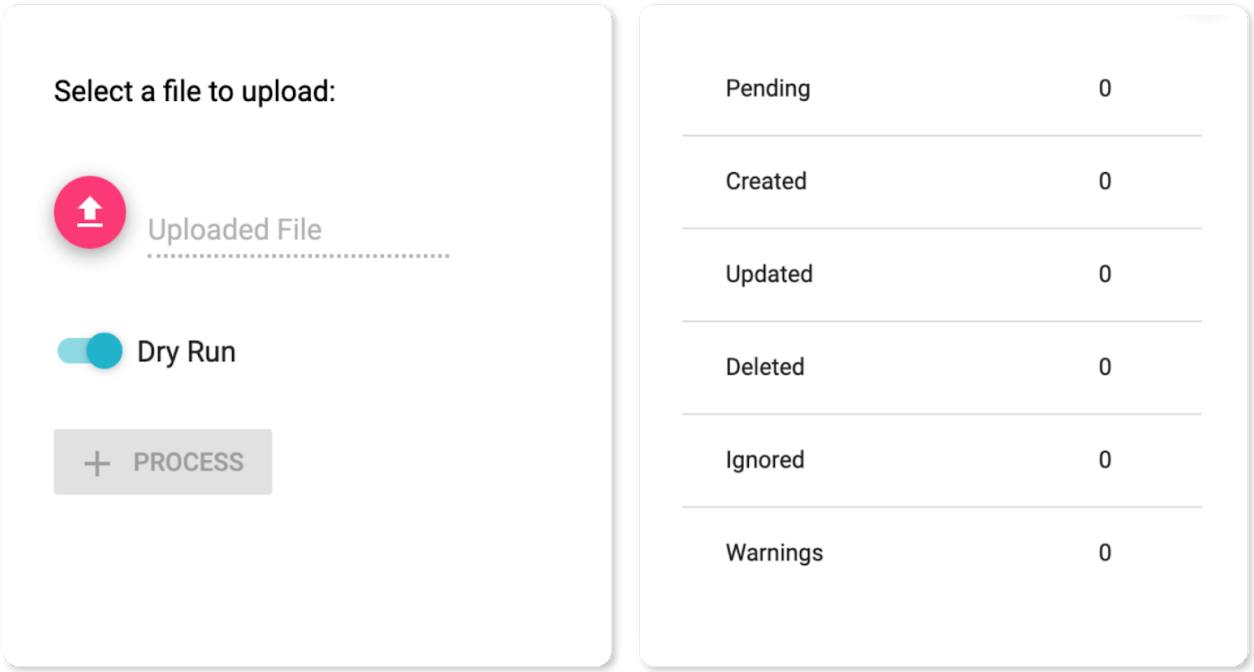
2. Once a file is selected, BIF will run a validation on that file as it loads it to the app and provides the user with a “pre-result” in the result layout. This checks if there are any errors or unexpected properties in the file before import. If warnings do exist, simply click on the word “warning” in the result layout and the log will display specific details about each error. Rectify these issues in your template, save your file and re-select it for re-upload.
3. The “Dry Run” function, on those modules that have it, acts much like the dry run in the DHIS2 import-export app as a validation process. It runs a check on the objects to be created, processing the proper alignment of the template, as well as the required syntax for each object. For example, the value type, “Integer_Zero_Or_Positive” not written in this format would return an error. By default, where present, the dry run button is turned on.
4. To execute the load, dry or real, click on the Process button. It will be “pink” when an accepted file is selected.

It is always advisable to do a “Dry Run” on the initial upload before the final upload to find out the validity of your import file and see how the system will receive it. Upon uploading, an analysis of the Dry Run will be displayed in the Results layout. If the analysis is successful, the number of objects created will be shown, and “Status ok” will be displayed at the bottom. If there are warnings or ignored, click on these to see the issue and rectify them in your import file. Note that a dry run does not create anything in the system.

5. Final upload. Ensure Dry Run is turned off, then click on the process button. This will create your object. It is always advisable to ensure the number imported corresponds to the expected number in your template, and that the objects did actually import as expected by navigating to the maintenance app.

Results layout

This layout outlines the analysis of the upload during the Dry Run and after upload.



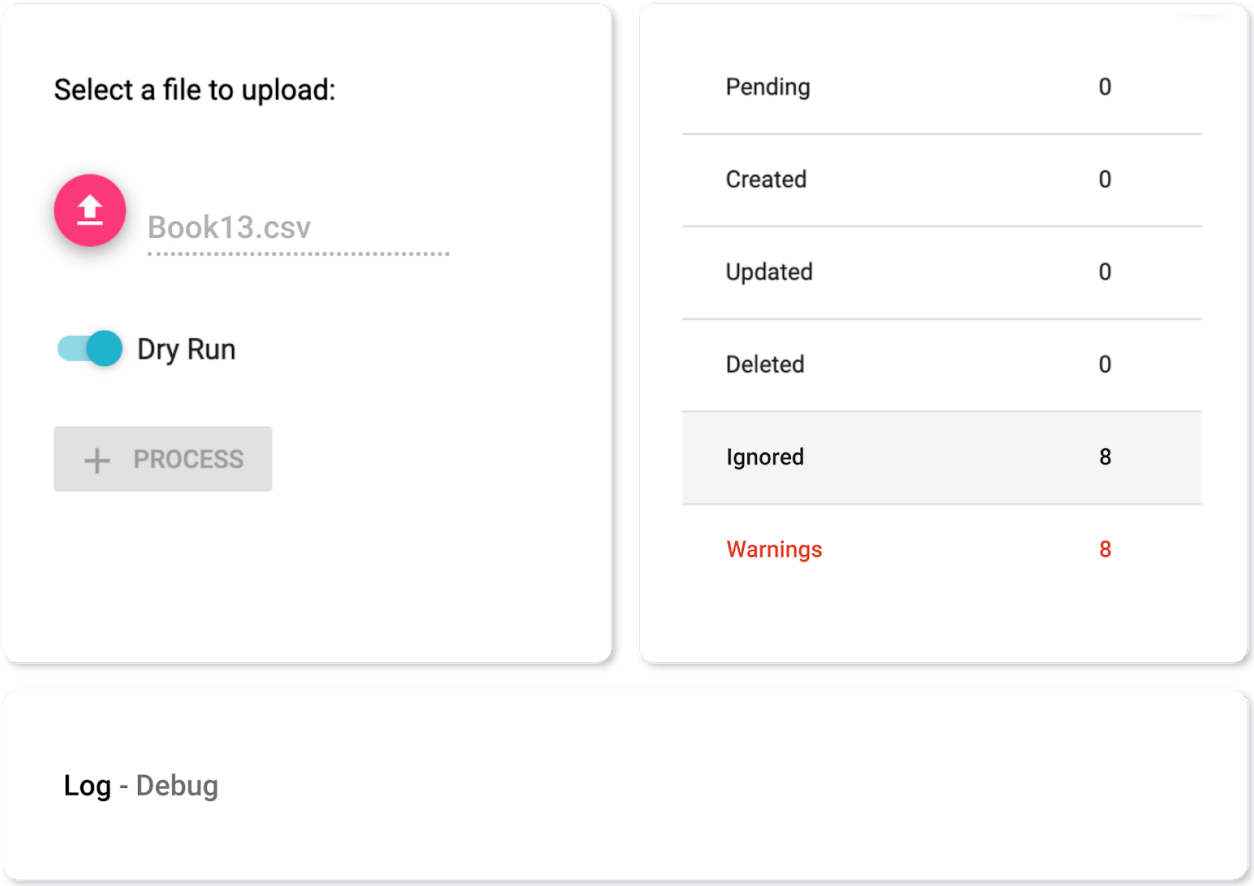
The analysis is broken down into the following sections:

- Pending: Shows the number of objects about to be created. This will populate an initial figure when the user file is selected for import, and decrease as import progresses.
- Created: Indicates the number of objects that have been created after import.
- Updated: Shows the number of objects existing in the system which have been updated. This is why specifying UIDs is useful in BIF templates, since it is possible to update objects.
- Deleted: Number of deleted objects.
- Ignored: Displays the number of objects that have been ignored, as a result of a violation within the BIF template.
- Warnings: Itemizes the number of violations or errors seen during the upload.

The logs

Log layout reports the description of violations or errors itemized in the result layout; the description serves as a notice for what needs reviewing or editing in the upload, or simply what was done.

The most common use of logs is to identify and fix errors. To do this: Click on “Warning(s)” in the results layout.



Then the description of the warnings is displayed in the Logs Layout. The information provided is useful to fix any problems with the BIF template. In the below error, for example, we are

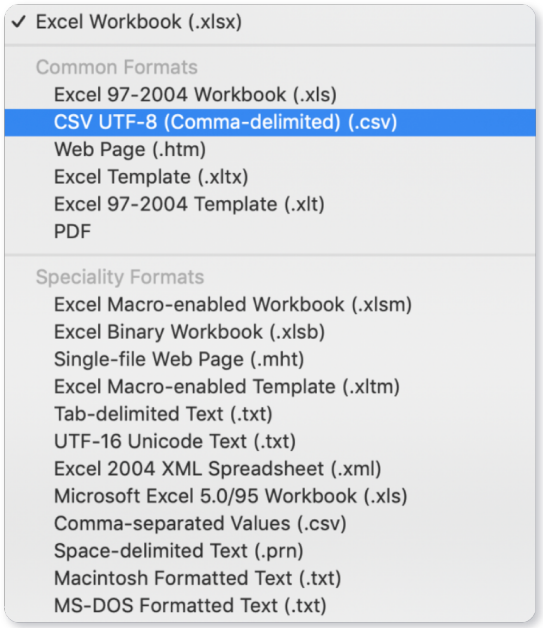
Log - Warnings

1. "Invalid date: undefined provided"
2. "Invalid date: undefined provided"
3. "Invalid date: undefined provided"
4. "Invalid date: undefined provided"
5. "Invalid date: undefined provided"
6. "Invalid date: undefined provided"
7. "Invalid date: undefined provided"
8. "Invalid date: undefined provided"

being told the date format needs to be changed.

Import format

For a successful file upload on BIF, files should be saved in “CSV UTF-8 (Comma delimited) (.csv)” file format. This is the recommended file format for the BIF template as it preserves special characters.



Helpful hints

Using dates: if your computer has a default date format that is not yyyy-mm-dd, then you will need to save your .csv file with this particular date format selected. Do not then reopen your .csv file as the date format will revert and your import will be unsuccessful. Import directly after saving with the appropriate date format.

Using special characters: be aware excel does not always handle special characters well. Check how your data has saved, perhaps in a text editor, prior to import. Special characters may have been deprecated and imported erroneously.

BIF has been built to accept the metadata of both PC and Mac .csv files. It is not configured, however, to receive .csv files where a ";" is used in place of a ",". This is common in .csv from french speaking countries. In these cases, the semicolon will need first replacing with a comma in a text editor.

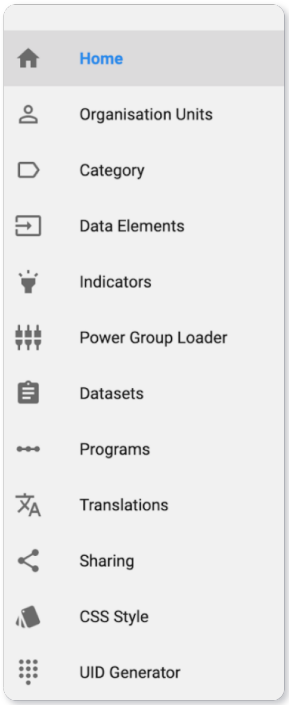
Configuration order

It is expected that users of BIF will have some idea about the various objects in DHIS2 and how they function together. To help with the configuration process, BIF is designed so that the order of objects in the left-hand menu reflects the order of configuration when building a new system from the start. Thus, objects with the least dependencies, or "smallest objects" are imported first, as these are necessary in turn to create objects with increasing dependencies or "larger objects".

For example, if the user wishes to create an aggregate data element disaggregated by male and female, it is first necessary to create the options for male and female, to group them into a category and then a category combination (cat combo), all before creating the data element that references the cat combo for male/female.

To successfully create metadata objects in DHIS2, the configuration order of the objects is crucial. The creation of some objects is dependent on the existence of other objects. The table below outlines this order.

Aggregate Domain	Order	Tracker/Events Domain	Order
Organisation Units & Groups	1	Organisation Units & Groups	1
Category options	1	Options & Option sets*	1
Categories	2	Data Elements	2
Category Combinations	3	Programs & Program stages	3
Date Elements	4	Program stage sections	4
Data Sets	5	Program rules & variables*	5
Sections	6	Program indicators*	6
Data element groups			A
Indicators			B
Indicator groups			C
Sharing			D
Translations			E



*objects marked with an asterisk are not yet available for import through BIF. See roadmap.

Troubleshooting

There are different ways to troubleshoot problems when using the BIF tool; here are three logs to check when attempting to resolve issues with an import when using the BIF app.

- BIF logs
- Browser console
- Server logs

Troubleshooting via BIF logs

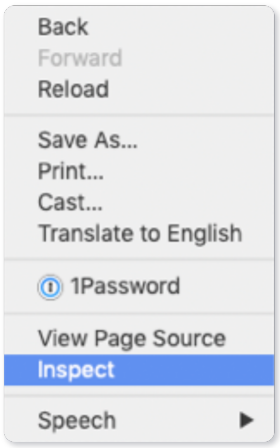
BIF logs provide user-friendly pointers to resolve issues. Click on Warnings/Ignored to see details of the warning/error logs as detailed above.

Troubleshooting via the Web Browser console

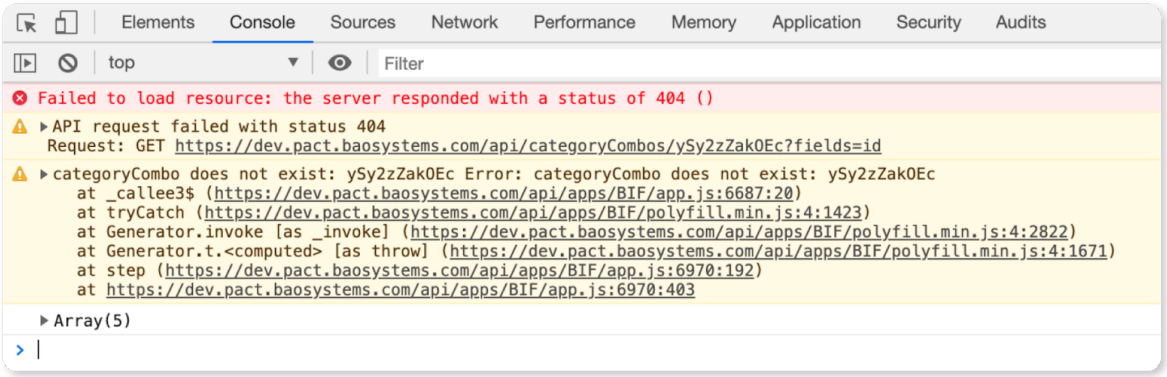
Not every possible error is explained by the logs. There are instances where it is necessary to monitor or inspect the browser console as this will record the process BIF was attempting and any error returned by the server. This process is slightly less “user-friendly” than the BIF logs, but still acts as a helpful guide.

Steps to inspect the browser console:

1. Right click on the page, then click on “Inspect” as shown below.



2. A window will appear in the user web browser that looks something like the image below. The most useful information on BIF processes can be found on the tab called “Console”. It can actually be helpful to have this console open when using BIF as any errors occurring will show up in real time for the user.



In the above example from a data element template, the error tells us that a category combo being referenced on a data element in the template does not exist on the target server. Therefore, you would either need to create it, or double check you are referencing the correct UID.

Troubleshooting via DHIS server logs

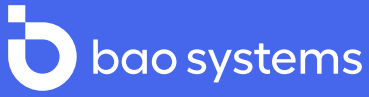
In rare cases, inspecting the server logs is required. This is because some processes do not register a warning in BIF or even in the web browser console. But if something is attempted on a server, every action is logged. By inspecting the server logs it will be possible to see any process attempted on the server, and any related warning or error it produced. For clients of BAO Systems, the server logs are accessible via the BAO Manager. The most important log for troubleshooting BIF is called “dhis.log”.

Roadmap

As DHIS2 develops there is always more scope to improve the BIF tool with extra properties and entirely new modules. In each release, we introduce innovative new features for BIF and there are many more to come in the future. This is what we are currently working on:

- Tracked Entity Attributes
- Option Sets
- Users
- User Groups
- Enrolment section forms

In addition, option groups, validation rules, validation rule groups, program indicators and program rules/variables are planned to be included further down the line.


bao systems