

# Developer Guide for Project HDF5\_BLS

Pierre Bouvet

February 27, 2025

Why are you here?

- [I want to import my data to the HDF5 file format](#)

## 1 Load data

Are you using a format that is already supported by the HDF5\_BLS package?

- [Yes but it doesn't work with my data.](#)
- [Yes but I would like to improve the code.](#)
- [No, I would like to add support for my data.](#)

### 1.1 Adding a user-specific function to an already supported format

You are in the situation where you are using a format that is already supported by the HDF5\_BLS package (for example ".dat") but that doesn't work with your data.

Here are the steps to follow:

1. Locate the python file that handles your data format in the `load_formats` folder of the HDF5\_BLS package. The name of the file should correspond to the name of the format you are using (for example "load\_dat.py" if you are using ".dat" files).
2. Add the function that will load your data to the file. The function should have the following signature:

```
1 def load_dat_Wien(filepath, parameters = None):
```

In the case where you don't need to load the data with parameters, the function should have the following signature:

```
1 def load_dat_Wien(filepath):
```

3. Write the code that will load your data. Your function should return a dictionary with at least two keys: "Data" and "Attributes". The "Data" key should contain

the data you are loading and the "Attributes" key should contain the attributes of the file. You can also add abscissa to your data if you want to, in that case, add the key "Abscissa\_ *name*" where *name* is the name you want to give to the abscissa (for example "Abscissa\_Time").

4. Go to the `load_data.py` file in the `HDF5_BLS` package and locate the function dedicated to the format you are using (for example "load\_dat\_file" if you are using ".dat" files)
5. Make sure that you are importing the function you just created:

```
1 from HDF5_BLS.load_formats.load_dat import load_dat_Wien
```

6. Then, define an identifier for your function (for example "Wien") and either create or add your identifier to the if-else statement. Don't forget to add your identifier to the "creator\_list" list in the "else" statement:

```
1 if creator == "GHOST": return load_dat_GHOST(filepath)
2 ...
3 elif creator == "Wien": return load_dat_Wien(filepath)
4 else:
5     creator_list = ["GHOST", "TimeDomain", "Wien"]
6     raise LoadError_creator(f"Unsupported creator {creator}, accepted
    values are: {'', '.join(creator_list))", creator_list)
```

7. Add a test to the function in the "tests/load\_data\_test.py" file with a test file placed in the "tests/test\_data" folder. This test is important as they are run automatically when the package is pushed to GitHub (ie: it makes my life easier i.
8. You can now use your data format with the `HDF5_BLS` package, and in particular, the GUI. You are invited to push your code to GitHub and create a pull request to the main repository :)

## 1.2 Improving an already supported function

You are in the situation where you want to improve a load function of the `HDF5_BLS` package (for example ".dat").

Here are the steps to follow:

1. Locate the python file that handles your data format in the `load_formats` folder of the `HDF5_BLS` package. The name of the file should correspond to the name of the format you are using (for example "load\_dat.py" if you are using ".dat" files).
2. Locate the function that loads your data. The function should have a name similar to (might not have parameters):

```
1 def load_dat_Wien(filepath, parameters = None):
```

3. Update the code. One good measure is to duplicate the function and comment one of the two versions. Then, write your code and run the tests. If the tests fail, you can always go back to the previous version. Note that if the test fails, the code cannot be pushed to GitHub.

4. If everything is sound, you can now use your new function with the `HDF5_BLS` package. You are invited to push your code to GitHub and create a pull request to the main repository :)
5. Note: If you want to improve the loading of the data to the hdf5 file (chunking for example), please contact the maintainer directly.

### 1.3 Adding a user-specific function to an already supported format

You are in the situation where you are using a new format that is not supported by the `HDF5_BLS` package.

Here are the steps to follow:

1. Navigate to the `load_formats` folder of the `HDF5_BLS` package.
2. Create a new python file with the name of the format you are using (for example `"load_unicorn.py"` if you are using `".unicorn"` files).
3. Add the function that will load your data to the file. The function should have the following signature:

```
1 def load_unicorn_Wien(filepath, parameters = None):
```

In the case where you don't need to load the data with parameters, the function should have the following signature:

```
1 def load_dat_Wien(filepath):
```

4. Write the code that will load your data. Your function should return a dictionary with at least two keys: `"Data"` and `"Attributes"`. The `"Data"` key should contain the data you are loading and the `"Attributes"` key should contain the attributes of the file. You can also add abscissa to your data if you want to, in that case, add the key `"Abscissa_name"` where `name` is the name you want to give to the abscissa (for example `"Abscissa_Time"`).
5. Go to the `load_data.py` file in the `HDF5_BLS` package and create the function dedicated to the format you are using (for example `"load_unicorn_file"` if you are using `".unicorn"` files)
6. Make sure that you are importing the function you just created:

```
1 from HDF5_BLS.load_formats.load_unicorn import load_unicorn_Wien
```

7. Add a test to the function in the `"tests/load_data_test.py"` file with a test file placed in the `"tests/test_data"` folder. This test is important as they are run automatically when the package is pushed to GitHub (ie: it makes my life easier i.
8. You can now use your data format with the `HDF5_BLS` package, and in particular, the GUI. You are invited to push your code to GitHub and create a pull request to the main repository :)

## Contact

For questions or suggestions, please contact the maintainer at:

[pierre.bouvet@meduniwien.ac.at](mailto:pierre.bouvet@meduniwien.ac.at).