

CEFDatabrowse Quick Start Guide

Nathan Scheirer
scheirer@iastate.edu

1 Installing

CEFDatabrowse has been extensively tested on Linux and Windows and is available for use with Python 2.7 (32 or 64 bit). The following Python dependencies are required by CEFDatabrowse, an attempt will be made to install these automatically on install but if for some reason one is not installed properly this list will be a reference for what you need:

- lxml
- numpy
- pillow
- qrcode
- cefpython3
- file-magic (python-magic-bin on Windows)
- matplotlib

All of these packages can be install via pip and are available for the platforms on which CEFDatabrowse has been tested. The following 3rd party libraries are used in various features of CEFDatabrowse but are not required:

- limatix-qautils (For experiment log and checklist viewing/editing)
- qautils (Provides a similar function as limatix-qautils)
- dataguzzler-python (For viewing .dgs data files)
- nditoolbox (Experimental NDT toolbox for viewing UT data on 3D models)

See the section below on configuration to learn how to install the 3rd party packages.

To install CEFDatabrowse simply open a command prompt and install via:

```
pip install databrowse
```

If this method does not work you can download the source code from the github repository [here](#). To install from source enter the Databrowse repository root directory and run:

```
python setup.py install
```

Development versions of Databrowse will be found on our Github as well as our issue tracker. If you encounter any bugs please report them in the issue section of our Github page so that we can fix the errors as quickly as possible. This project is also licensed as open source so submitting contributions to our project is welcome.

2 Configuration

Following a successful installation of CEFDatabrowse a few configuration variables will need to be changed in order for the software to work properly. To start out, create a new directory in which you will want to serve any files through Databrowse. We recommend that in your root directory you create a folder called “databrowse.” This will keep your data organized into one central location. Therefore, with a folder called “databrowse” in your root directory you will need to create a folder within that called SOPs. This directory contains all of your Standard Operating Procedures and checklists. CEFDatabrowse will be able to reference the files located in this folder from anywhere and create new checklists or files whenever you need them. Now, the following is an example of a CEFDatabrowse configuration file which can be opened by running “databrowse -e” in the command prompt or by opening the “settings” tab in the “file” menu of CEFDatabrowse:

```
[databrowse]
width = 1040
height = 702
x = 0
y = 0
dataroot = C:/databrowse
limatix-qautils = C:/opt/databrowse-utils/limatix-qautils
qautils = C:/opt/databrowse-utils/QAutils

[3rdparty]
dataguzzlerlib = C:/opt/dataguzzler-lib/python
nditoolbox = C:/opt/NDITool
```

The values “width”, “height”, “x”, and “y” will not need to be changed, these are merely values saved by CEFDatabrowse to determine screen size and position. The important variable is “dataroot,” this is the directory from which Databrowse can serve files. Anything above or outside this folder will return as an access violation. Therefore, make sure that the files you will want to serve

are under this directory and be aware that Databrowse isnt intended to have this value change often. Once this is set then there shouldnt be much of a reason to change it again. The next variables, “limatix-qautils“ and “qautils“ perform very similar functions and Databrowse will operate just fine with just “limatix-qautils“ installed. To inform Databrowse of how to use these programs, simply replace the string following the variable with the path to the source code for the program. This is how anything listed under the 3rdparty tag will be associated as well. CEFDatabrowse will add these folders to its path in order to find the correct scripts to run. Therefore, any 3rd part program you wish to use in Databrowse can be listed under the 3rdparty tag and if there is functionality in Databrowse that supports that program then it will automatically be included. This should conclude the required configuration for CEFDatabrowse to run as intended. If you need more assistance you can open the manual from the “file“ tab located at the top of the CEFDatabrowse window.

3 Tips and Tricks

1. Specific handlers and styles can be called from the navigation bar by appending variables to the end, i.e. to create a specimen database, you will need to move the “src“ folder inside the “example_templates“ folder which is located in “databrowse/databrowse/plugins/db_specimen_database“ within the Databrowse source files and then navigate to a url like this:
“/databrowse/specimens?handler=db_specimen_database&style_mode=add_specimen“
2. The “file“ menu on the top bar of the CEFDatabrowse window contains a lot of helpful links such as, a link to the “settings“ file, a link to the manual, and a button to close the app.
3. In Windows, when CEFDatabrowse is installed a .exe file will be created in the “Scripts“ folder of your Python installation. You can then create a shortcut to this file and then you can open CEFDatabrowse from an icon on your desktop. You can find your Python installation path by typing this into a command prompt: “where python“