

Feedback EDL Generator

1. [#Setup](#)
 - a. [#Environment](#)
 - b. [#Dependencies](#)
 - c. [#Run](#)
2. [#Development](#)
 - a. [#Overview](#)
 - b. [#Edit Template](#)
 - c. [#Modules and Functions](#)
 - i. [#Major Modules](#)
 - ii. [#EPICS](#)
 - iii. [#GUI](#)
 - iv. [#Raw Templates](#)
 - v. [#Compiled Templates](#)
 - vi. [#Other](#)
3. [#Release](#)

Glossary

- Feedback_id is a string of the following format FBXY:TRXY (e.g. FB03:TR04)

Setup

Environment

- You can develop the Fbck EDL generator in both, development (e.g. lcls-dev2) and production (lcls-builder) environments
 - Some PVs may not be accessible on development
 - Make sure you have the proper Python setup (e.g., at least, version 2.6)
 - If you don't, ask Jingchen for help

Dependencies

- [EDL Generator](#) (current version)
- \$TOOLS/python/python2.6.4/external-packages/Cheetah (at least version 2.4)
- \$TOOLS/python/python2.6.4/external-packages/epics (at least version 3)
- Many libraries that come with Python 2.6

Run

- You can run Fbck EDL generator scripts from the command-line
- Check out the tools/script module and keep only the files/directories that start with "fbck"

```
cvs co -d . tools/script/  
shopt -s extglob  
rm -rf !(CVS| *cvs* |fbck*) #We only work on the Fbck EDL generator scripts
```

- Run the main FBCK EDL generator script to see the command line options

fbckedlgen.py

Usage: fbckedlgen.py -g [other options] feedback_id beamline_name|'plots'|"offsets'

Following beamline names are supported:

bsy, dl2_a, gun, injector, l2, l3, li28, ltu, und, x_cavity

Options:

--version	show program's version number and exit
-h, --help	show this help message and exit
-d, --display_edm	display the pregenerated EDM screens (applies only if -g is specified, too)
-g, --generate_all	generate all screens at once (no on-the-fly generation)
-i, --ignore_used	ignore USED PVs
-o OUTPUT_DIR, --output_dir=OUTPUT_DIR	specify the ABSOLUTE directory path for the generated edl

Deprecated Options:

These options are useful for on-the-fly generation of EDM screens only.

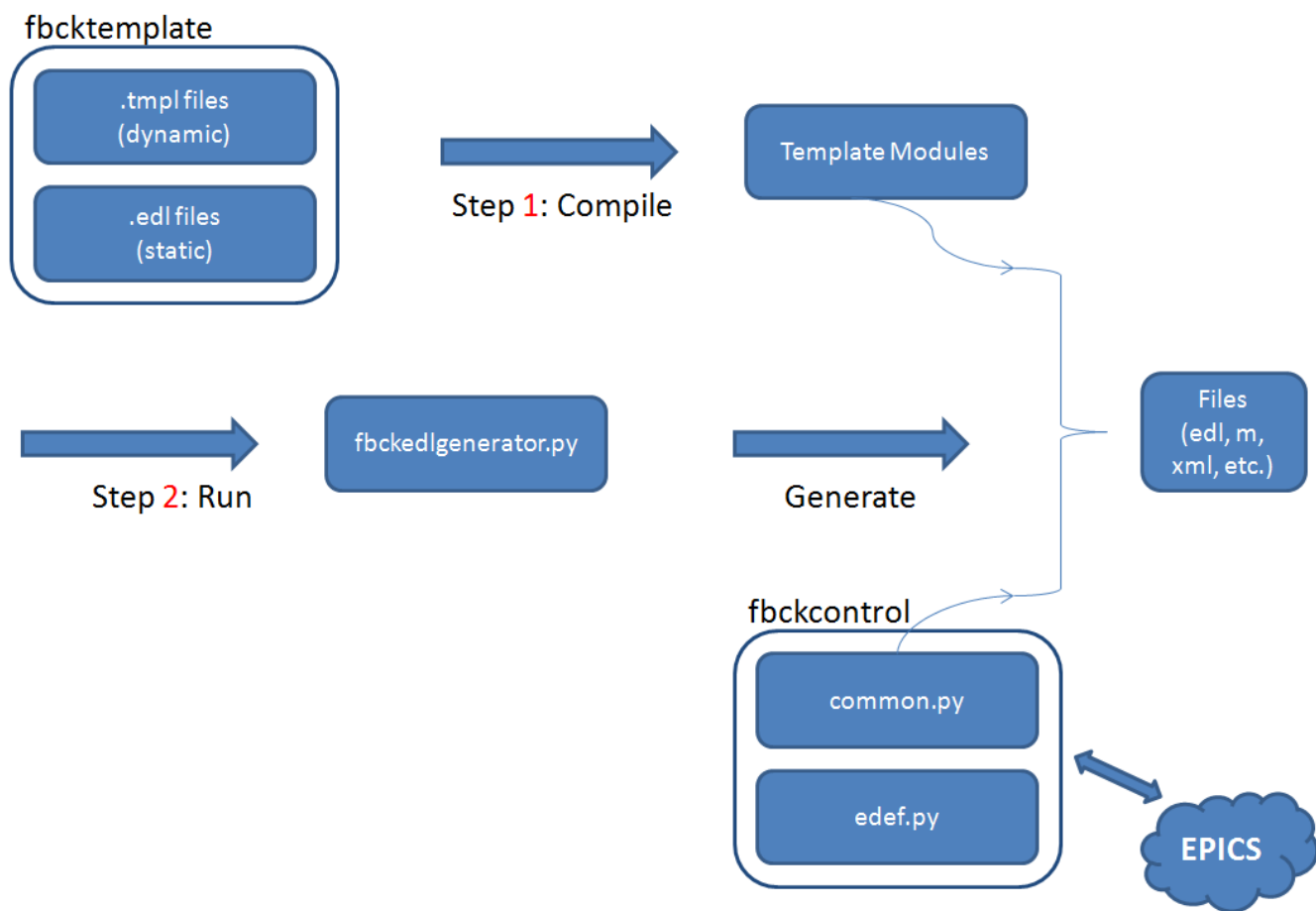
-a A_PAGE, --actuator_page=A_PAGE	select the specified actuators page
-f OFFSETS_INDEX, --offsets_index=OFFSETS_INDEX	set the index for the state offsets dialog
-m M_PAGE, --measurement_page=M_PAGE	select the specified measurements page
-p POI_INDEX, --poi_index=POI_INDEX	set the index of the POI to display the data from (0 = all)
-s S_PAGE, --state_page=S_PAGE	select the specified states page

Here is an example of a valid command that generates an EDL file in the current directory:

```
fbckedlgen.py -gd FB03:TR04 gun
```

Development

Overview



There is also a design document: [EDL Generator design.doc](#)

Edit Template

- The raw feedback templates are stored in .tmpl files under \$STOP/fbcktemplate
 - Some templates use static .edl files from the same directory; you can edit these auxiliary files with EDM directly after you add an [EDL header](#) to the top
 - Don't forget to remove the header after you save!
- After you finish editing a feedback template, you must compile it into a Python module and move the resulting Python script to the \$STOP directory, e.g.

```
$TOOLS/python/python2.6.4/external-packages/Cheetah/bin/cheetah compile fbcktemplate/fbckdetails.tmpl
mv fbcktemplate/fbckdetails.py .
```

Modules and Functions

Major Modules

fbckconfig.py

- Contains the repository of parameters and values that are used in the feedback templates

fbckedlgen.py

- An entry point to the application (see [#Run](#))
- Parses command-line options
- Creates a Cheetah namespace, see http://www.cheetahtemplate.org/docs/users_guide_html/
- Generates files from a corresponding template as selected via command-line arguments

fbckgenerateall.py

- A utility that generates all feedback details EDM screens (configurable inside)

fbcktemplate.py

- Super class for all feedback templates
- Keeps track of used EDM colors (for plots)
- Provided support for indexed variables and used indices
- Implements pagination

EPICS

fbckcontrol/common.py

- Contains all-purposed function for getting/setting PV values

fbckcontrol/edef.py

- Contains function for acquiring/releasing EDEFs

GUI

- There are simple Swing GUIs under \$TOP/fbckjython that are used by [#fbckregold.py](#)
- To build gui.jar

```
cd $TOP/fbckjython
sh build.sh
```

Raw Templates

- Raw Cheetah templates are located under \$TOP/fbcktemplate
 - .edl files contain some static EDM widgets (e.g. beamlines)
 - .tmpl files are the dynamic template files, similar to JSPs
- You must compile these templates before using them!
- See http://www.cheetahtemplate.org/docs/users_guide_html/ for more details

Compiled Templates

fbckavconfig.py

- Template class for an av configuration file

fbckdetails.py

- Template class for an EDM screen that contains details about a feedback

fbckmatlabexport.py

- Template class for a Matlab script that exports the feedback data to a .mat file

fbckoffsets.py

- Template class for an EDM screen that displays information about feedback offsets

fbckplots.py

- Template class for an EDM screen that displays feedback plots

Other

fbckaccessor.py

- Provides functions for getting values of various types of feedback parameters that are used in templates

fbckmonitor.py

- Utility for monitoring usage of feedbacks
- To start the corresponding daemon, you have to be on lcls-daemon2 as laci

```
/etc/rc3.d/S99st.fbckmonitor start
```

To stop the corresponding daemon

```
/etc/rc3.d/S99st.fbckmonitor stop
```

- Error messages are logged to /u1/lcls/tools/fbckMonitor/fbckmon.log

fbckregold.py

- Refreshes a gold orbit
- Calls Swing GUIs that are under \$TOP/fbckjython

fbckresetoffsets.py

- Resets feedback offsets

fbckrestoreacts.py

- Restore feedback actuators

fbckrunav

- A bash script to run the ArchiveViewer with the proper AV configuration file (see also [#fbckavconfig.py](#))

fbckupdateactrefs.py

- Updated feedback actuator references

Release

- Please, release in both, the development (e.g. lcls-dev2) and production (e.g. lcls-builder) environments!
- Commit the changes to CVS
- Go to \$TOOLS/script/fbckcontrol/
- Run CVS update

```
cvs update
```

- Go to \$TOOLS/script/fbckjython/
- Run CVS update

```
cvs update
```

- Go to \$TOOLS/script/fbcktemplate/
- Run CVS update

```
cvs update
```

- Go to \$TOOLS/script
- Run CVS update **on fbck scripts only!**

```
cvs update fbckaccessor.py fbckavconfig.py fbckconfig.py fbckdetails.py fbckedlgen.py fbckgenerateall.py  
fbckmatlabexport.py fbckmonitor.py fbckoffsets.py fbckplots.py fbckregold.py fbckresetoffsets.py  
fbckrestoreacts.py fbckrunav fbcktemplate.py fbckupdateactrefs.py
```

- Note: If nobody else is using the prefix fbck*, you can also do cvs update fbck*