Python EPICS Generation

Several python programs exist for reading tabbed text spreadsheets into a python object so the spreadsheet can be searched , sorted and edited with python methods. Also python classes have been written to read/write epics databases (,db or template files). The PVs are stored as python objects and PVs can be edited or built up with python methods. The template file can be written with python methods. A similar python class has been written to handle Control System Studio screens, where a screen is represendted as a python object, with each widget on the screen also as python objects. The class reads/writes XML so CSS can read the files. There also exists a python function to read EDM screen files to generate CSS screens., allowing the conversion of EDM to CSS.

Location of python scripts

https://svn.inside.anl.gov/repos/dgs/DGS_SW/pyScripts/trunk/pyScripts/

On DGS1 the working copy to use is at

/home/dgs/tmadden/swWork/pyScripts

To get the python scripts:

On a Cygwin or unix prompt:

```
#get a working copy of the scripts
```

```
svn checkout
https://svn.inside.anl.gov/repos/dgs/DGS_SW/pyScripts/trunk/pyScripts/
./pyScripts

#open editor to see highest level py script

nedit genSystem.py

#run python interpreter in interactive mode

python

#in the python interpreter, load in the main py script.

execfile('genSystem.py')
```

You will see a simple help screen.

What python generates and where it is stored

The python scripts generate

1. Screens as .opi files.
2. EPICS database template files.
3. EPICS database db files.
4. .c and .h C source files.
5. IOC startup scripts, st.cmd's
6. cdCommand files

The files are stored in two places:

1. The current directory where the python scripts run from.
2. /home/dgs/CSS-Workspaces/Default/CSS for opi screen files.
3. /global/develbuild/gretTop/9-22/dgsDrivers/dgsDriverApp/src for .c and .h files.
4. /global/develbuild/gretTop/9-22/dgsDrivers/dgsDriverApp/Db for .db and .template files.
5. /global/develbuild/gretTop/9-22/dgsDrivers/dgsIoc/iocBoot/iocArray for st.cmd and cdCommand files.

Note that /global/develbuild is set in code_location global variable, so it may be /global/devel.

For DGS digitizer the following files are created:

- dgsDigRegisters.template
- digitizer3.opi
- vme01.DGS.cmd
- digEngChanRegisters.opi
- cdCommands_DGS
- vme05.DGS.cmd
- vme04.DGS.cmd
- vme03.DGS.cmd
- vme02.DGS.cmd
- vme10.DGS.cmd
- vme09.DGS.cmd
- vme08.DGS.cmd
- vme07.DGS.cmd
- vme06.DGS.cmd
- vme11.DGS.cmd
- asynDigParams.h
- asynDigParams.c
- dgsDigUser.template
- Digitizer.opi
- segment.opi
- dgsGlobals_DGS_VME09.db
- dgsGlobals_DGS_VME08.db
- dgsGlobals_DGS_VME05.db
- dgsGlobals_DGS_VME04.db
- dgsGlobals_DGS_VME07.db
- dgsGlobals_DGS_VME06.db
- dgsGlobals_DGS_VME01.db
- dgsGlobals_DGS_VME03.db
- dgsGlobals_DGS_VME02.db
- dgsGlobals_DGS_GLBL.db
- dgsGlobals_DGS_VME10.db
- dgsGlobals_DGS_VME11.db
- DGTL_Global.opi

For DGS Trigger

For Clover Digitizer the following files are created

- vme01.CLO.cmd
- vme02.CLO.cmd
- vme03.CLO.cmd
- vme04.CLO.cmd
- cdCommands_CLO
- vme05.CLO.cmd
- dgsGlobals_CLO_VME05.db
- dgsGlobals_CLO_VME04.db
- dgsGlobals_CLO_VME01.db

- dgsGlobals_CLO_VME03.db
- dgsGlobals_CLO_VME02.db
- dgsGlobals_CLO_GLBL.db

When to run make

You must run make when generating files for DGS. For DFMA and clover, you first generate drivers/pvs for DGS, make, then check into svn. To generate for clover/dfma, you svn update the gretTop/9-22 area, then run the py scripts. No make is run for dfma or clover.

How to open spreadsheet in Linux

Get the latest version of the spreadsheet from the svn repo.

From UNIX prompt start OpenOffice.

soffice &

You can open the spreadsheet.

SaveAS your spread sheet as a Text file, delimited by Tabs.

In OpenOffice

- File->SaveAs
- Pick filename (MDRM.csv,MTRM.csv), and below, scroll to Text
- Continue when prompted to save in this format.
- Pick the delimiter as Tab. Leave text delimiter as “.
- File->Open your original spreadsheet.
- File->close the text spreadsheet.

How to generate DGS system- Digitizers

- Log onto dgs1, because you must run make eventually.

- cd /home/dgs1/tmadden/swWork/pyScripts
- Start up python shell in fresh xterm.
    - xterm &
    - In xterm, type python
- In python load scripts by typing:
    - execfile('genSystem.py')
- Set global variables
    - code_location='/global/devel' or '/global/develbuild'
    - code_location = None if you are not copying files into build area, say for testing.
        - code_location defaults to '/global/devel'
    - is_do_screens = 1 to generate screens, or 0 for no screen generation
        - default is 1
    - ss_delimiter = '\t\' or ',' for tab or comma delimited text spreadsheet.
        - Default is '\t' for tabbed spreadsheets
- Run the script
    - *genDGSDigitizerDriver('[csv pathname/filename]')*
- If you had copied files to build area you must run make
    - ssh con5
    - cd /global/devel/gretTop/9-22/dgsDrivers
    - make
    - cd ..
    - cd dgsIoc
    - make
- Set your IOCs to boot from the build location
- Reboot IOCs

Python script for generating DGS system. Copy this into Python.

```
execfile('genSystem.py')

code_location='/global/devel'

ss_delimiter='\t'

is_do_screens=0

genDGSDigitizerDriver('MDRM.csv')
```

Procedure for Configuring DSSD using Python

To generate DSSD files, log onto dgs1. You generate the files then check into svn. On darak you checkout and you are done.

On DGS, get latest spreadsheet and save as a text spread sheet as in above. In python run this script

```
execfile('genSystem.py')

code_location='/global/devel'

ss_delimiter='\t'

is_do_screens=0

genDFMADig('MDRM.csv')

genDFMATrig()
```

Now from a unix terminal, on dgs1,

```
cd /global/devel/gretTop/9-22/dgsIoc

svn commit –m “python gen for dfma”
```

Now you log into DFMA

To log into clover from sonata or dgs1, ssh –Y dgs@nat1b.phy.anl.gov

Procedure for Configuring Clover using Python

To generate clover files, log onto dgs1. You generate the files then check into svn. On clover you checkout and you are done.

On DGS, get latest spreadsheet and save as a text spread sheet as in above. In python run this script

```
execfile('genSystem.py')

code_location='/global/devel'

ss_delimiter='\t'

is_do_screens=0

genCloDig('MDRM.csv')

genCLOTrig()
```

Now from a unix terminal, on dgs1,

```
cd /global/devel/gretTop/9-22/dgsIoc

svn commit –m “python gen for clover”
```

Now you log into Clover

To log into clover from sonata or dgs1, ssh –Y dgs@nat1b.phy.anl.gov

Cd /global/devel/gretTop/9-22

svn update

When Python Scripts Generate Errors

This is usually caused by errors in spread. The python code will print out the spreadsheet line and a stack trace dump. You will have to know something about python to understand this stack trace. The most common errors are

1. Setting t_delimiter incorrectly, the spreadsheet col delimiter.
2. The bit field subdiscriptor has an error, say, a colon instead of semicolon.
3. Commas in description column in a comma delimited spreadsheet.
4. Tabs in columns in a tab-delimited spreadsheet.

Important Functions, Variables, Commands

code_location

is_do_screen

ss_delimiter

help_screen

genDGSMasterTrigger()

genDGSRouterTrigger()

genDGSDigitizerDriver(ssname)

What each python file is

genSystem.py

This is the master python file that loads everything and has top level functions that generate databases and drivers.

epicsclass.py

This file is a class that represents a PV. An epics database is stored in python as a pyton list of epicspv objects.

epicsParse.py

epicsParse is a collection of functions for parsing databases and generating databases. EPICS databases can be read from template files, db files, or created from spreadsheets. The databases are represented as python lists of epicspv objects. epicsParse has functions to load, write epics databases, search/replace in databases, and generate databases.

genCode.py

Collection of functions for generating C code and st.cmd files.

guiClass.py

This py file contains a class to represent a Control SystemStudio (CSS) screen, and widgets.

screenMaker.py

Class that has methods for generating and search/replacing in CSS screens.

spreadsheet.py

Class to represent a spreadsheet. Objects contain all the rows and cols of the spreadsheet. Can read/write/parse spreadsheets.

ssAnalyze.py

Code for analyzing epics databases and spreadsheets. Used for generating a spreadsheet when given an epics database.

parseopi.py

Has two functions: a deprecated function to convert epics databases into opi files. Not used anymore, as screenMaker.py works better. There is another function edm2opi that converts EDM screens to CSS screens. This is useful.

Parse.opi

Functions for converting Carlware Gretina databases to newer asynDriver based databases. Older code not used anymore.

udpRcv.py

A simple UDP receiver to test the onMonotor sender of DGS. Has nothing to do with generating databases.

Createfiles.py

Deprecated.