```
run_cmdline script

*Run a single interactive job using command line.

*Format: run_cmdline <full_path_to_rc_file>

#!/bin/sh
export TUT_COMP_ROOT=/home/elwasif/CCA/tutorial/src/sidl
export CCAFE_HOME=/home/elwasif/CCA/ccaffeine

# setup paths
SIDL_DLL_PATH="" ;DL_PATH=/home/elwasif/babel7.4/lib
for i in `grep '^\s*path' "$1" | awk '{print $3}'`; do
DL_PATH=$DL_PATH:$;;
done
export LD_LIBRARY_PATH=$DL_PATH
export SIDL_DLL_PATH=:echo $DL_PATH | sed 's/:/;/g'`
$CCAFE_HOME/cxx/dc/user_iface/ccafe-single --ccafe-rc $1
```

```
run_gui script

*Run a single interactive job using the GUI

*Format: run_gui <full_path_to_gui_rc_file>

#!/bin/sh
export TUT_COMP_ROOT=/home/elwasif/CCA/tutorial/src/sidl
export CCAFE_HOME=/home/elwasif/CCA/ccaffeine

#setup paths (same as in run_cmdline)
... ... ... ...

$CCAFE_HOME/cxx/dc/user_iface/ccafe-single --type server
--port 3314 --ccafe-rc $1 &
sleep 2
$CCAFE_HOME/bin/rungUI --builderFort 3314
exit 0
```

Ccaffeine Convenience Script

```
run_gui_parallel script (1)

#!/bin/sh -f
# Format: run_gui_parallel <num_proc> <full_path_to_gui_rc_file>
# Configuration stuff
export mpirun_vsr/local/bin/mpirun
export CLASSPATH=$CCAFE_HOME/java:$CLASSIC_CCA_ROOT/java
export gui=$CCAFE_HOME/jin/rungUI
export TUT_COMP_ROOT='pwd'

export javaopts=" -Djava.compiler=NONE -classpath $CLASSPATH"
export procfile="/tmp/processors.$$"
export machfile="/tmp/processors.$$"
export machfile="/tmp/machines.$$"

# This tells CCAFFEINE to put the frameworks output streams
# into the current directory, into files named poutN and
# pErrN, where N is the process number (starting from 0).
echo "Look for application output in pout[01] and pErr[01] in
this directory"
```

```
run_gui_parallel script (2)

# Create a 'processors' file to tell the framework where to
# find itself and the GUI.
echo 127.0.0.1 server > $procfile
i=0
while [$i-lt$1]
do
echo $i client >> $procfile
i='expr $i+1'
done
# Create the mpirun machines file
echo 127.0.0.1 > $machfile
# Start the GUI and wait briefly to give it a chance to
# initialize
echo Launching multiplexer...
java $javaopts \
gov.sandia.ccaffeine.dc.distributed.MuxingProcess \
--name 127.0.0.1 --timeout 0 --file $procfile &
sleep 3
```

```
#Generate the setup_paths script

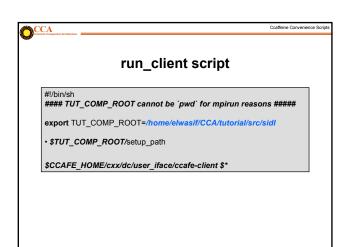
SIDL_DLL_PATH=""
DL_PATH=$BABEL_ROOT/lib
for i in 'grep '\\s*path' "$2" | awk '{print $3}'\; do
DL_PATH=$DL_PATH=$i;
done
export SIDL_DLL_PATH=`echo $DL_PATH | sed 's/:/;g'\

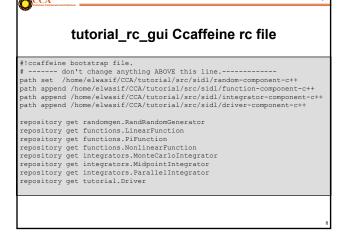
# Store commands into setup_paths file (to be included into
# scripts at target mpi nodes)
cat > $TUT_COMP_ROOT/setup_paths << EOF
export LD_LIBRARY_PATH=$DL_PATH
export SIDL_DLL_PATH="$SIDL_DLL_PATH"
EOF
```

```
run_gui_parallel script (3)

# Launch the framework
echo Launching framework...
%mpirun -np %1 -machinefile %machfile %TUT_COMP_ROOT/run_client\
--file %proofile --ccafe-rc %2 &
sleep 5
# Launch GUI
echo Launching GUI...
%gui

# Look for any stray files or processes
echo Cleaning up
rm -f %proofile %machfile
# This may be overkill
killall ccafe-client\
gov.sandia.ccaffeine.dc.distributed.MuxingProcess \
runGUI gov.sandia.ccaffeine.dc.user_iface.BuilderClient java
```





```
run_cmdline_python file

#!/bin/sh
export TUT_COMP_ROOT=/home/elwasif/CCA/tutorial/src/sidl
export CCAFE_HOWE=/home/elwasif/CCA/ccaffeine
export_CAFE_HOWE=/home/elwasif/babel-0.8.4

PYTHON_VERSION=2.2
# setup paths
export_SIDL_DEBUG_DLOPEN=1
SIDL_DLL_PATH=""
PYTHONLIB_PATH=/usr/lib/python2.2/config

DL_PATH=$PYTHONLIB_PATH:$BABREL_ROOT/lib
for i in `grep '^\s*path' "$1" | awk '{print $3}'`; do
DL_PATH=$DL_PATH:$!;
done
if [ ! $?LD_LIBRARY_PATH | ; then
export_LD_LIBRARY_PATH=$DL_PATH
else
export_LD_LIBRARY_PATH=$DL_PATH | sed 's/:/;g'`
```

