

## run\_cmdline script

```
#!/bin/sh

export TUT_COMP_ROOT=`pwd`

DL_PATH=$BABEL_ROOT/lib
for i in `ls $TUT_COMP_ROOT | grep component`; do
  DL_PATH=$DL_PATH:$TUT_COMP_ROOT/$i;
done

export LD_LIBRARY_PATH=$DL_PATH
export SIDL_DLL_PATH=`echo $LD_LIBRARY_PATH | sed 's/:;/g'`

$CCAFE_HOME/cxx/dc/user_iface/ccafe-single --ccafe-rc $1
```

1

## run\_gui script

```
#!/bin/sh

export TUT_COMP_ROOT=`pwd`

DL_PATH=$BABEL_ROOT/lib
for i in `ls $TUT_COMP_ROOT | grep component`; do
  DL_PATH=$DL_PATH:$TUT_COMP_ROOT/$i;
done

export LD_LIBRARY_PATH=$DL_PATH
export SIDL_DLL_PATH=`echo $LD_LIBRARY_PATH | sed 's/:;/g'`

CURDIR=`pwd`
$CCAFE_HOME/cxx/dc/user_iface/ccafe-single --type server --port 3314 \
--ccafe-rc $CURDIR/$1 &
sleep 2
$CCAFE_HOME/bin/runGUI --builderPort 3314
```

2

## run\_gui\_parallel script (1)

```
#!/bin/sh -f
# Format run_gui_parallel num_proc full_path_to_gui_rc_file
# Configuration stuff
export mpirun=/usr/local/bin/mpirun
export CLASSPATH=$CCAFE_HOME/java:$CLASSIC_CCA_ROOT/java
export gui=$CCAFE_HOME/bin/runGUI

export javaopts="-Djava.compiler=NONE -classpath $CLASSPATH"
export CCAFFEINE_OUT_DIR=`pwd`
export procfile="/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"
```

3

## 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.caffeine.dc.distributed.MuxingProcess \
    --name 127.0.0.1 --timeout 0 --file $procfile &
sleep 3
```

4

## run\_gui\_parallel script (3)

```
# Launch the framework
echo Launching framework...
$mpirun -np $1 -machinefile $machfile `pwd`/run_client \
        --file $procfile --ccafe-rc $2 &
sleep 5
# Launch GUI
echo Launching GUI...
$gui

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

5

## run\_client script

```
#!/bin/sh

export TUT_COMP_ROOT=/home/elwasif/CCA/tutorial/src/sidl

DL_PATH=$BABEL_ROOT/lib
for i in `ls $TUT_COMP_ROOT | grep component`; do
    DL_PATH=$DL_PATH:$TUT_COMP_ROOT/$i;
done

export LD_LIBRARY_PATH=$DL_PATH
export SIDL_DLL_PATH=`echo $LD_LIBRARY_PATH | sed 's/:/;/g'`

$CCAFE_HOME/cxx/dc/user_iface/ccafe-client $*
```

6

## tutorial\_rc\_gui Ccaffeine rc file

```
#!/ccaffeine bootstrap file.  
# ----- don't change anything ABOVE this line.-----  
path set /home/elwasif/CCA/tutorial/src/sidl/random-component-c++  
path append /home/elwasif/CCA/tutorial/src/sidl/function-component-c++  
path append /home/elwasif/CCA/tutorial/src/sidl/integrator-component-c++  
path append /home/elwasif/CCA/tutorial/src/sidl/driver-component-c++  
  
repository get randomgen.RandRandomGenerator  
repository get functions.LinearFunction  
repository get functions.PiFunction  
repository get functions.NonlinearFunction  
repository get integrators.MonteCarloIntegrator  
repository get integrators.MidpointIntegrator  
repository get integrators.ParallelIntegrator  
repository get tutorial.Driver
```