Commands and Scripts to Execute Simcodes from within iSIGHT

Revised by Max Lai (ylai@sightna.com) in 2003 for such issues:
1. ProE 2. Ansys

March 1, 2002

This document describes the commandlines and scripts you would need to execute some simcodes from within iSIGHT. It is mainly based on Kevin Harris Commands/scripts list you can get from his web site.

If you find this useful and have some additions you want see, please mail them to Holger.Wenzel@engineous.com.

Contents

1 Commercial Simcodes 3

1.1 Abaqus 3

1.2 Adams 3

1.3 Ansys 3

1.3.1 Unix 3

1.3.2 Windows 4

1.4 Autoform 4

1.5 CAEFEM 4

1.6 CFX5 Solver 4

1.7 DADS 4

1.8 Excel 5

1.9 FEMAP 5

1.10 Fluent 5

1.10.1 Unix 5

1.10.2 Windows 6

1.11 Gambit 6
1.12 GT-Power 6
1.12.1 Unix 6
1.12.2 Windows 6
1.13 Hydsim 6
1.14 ICEM CFD 6
1.15 IDEAS 7
1.16 LSDYNA 7
1.17 Matlab 7
1.18 matrixX 7
1.19 Maze and dyna 2d 7
1.20 MoldFlow 8
1.21 Nastran 8
1.21.1 Unix 8
1.21.2 Windows 8
1.22 Ncode 8
1.23 PAMCRASH 9
1.24 Patran 9
1.25 Pro/E 9
1.26 RADIOSS 9
1.27 SYSNOISE 9
1.28 Tycon 10
1.29 UniGraphics 10

2 Generic Scripts 10
2.1 C-Shell 10
2.2 ksh Script 11
2.3 Perl 12
2.4 Remote Shell 13

1 Commercial Simcodes

1.1 Abaqus

Script to check completion

```
/apps/ABAQUS-57/abaqus job=deck.inp user=mounts &
until (grep "Normal termination to simulation code." deck.log )
do
   sleep 2
done
../shockspect3d.x
cp abaqus.rpt ../abaqus1.rpt
cp deck.spc ../deck1.spc
rm deck.dat deck.msg deck.log deck.sta deck.fil deck.com deck.spc
echo "ABAQUS and ShockSpectrum analyses completed. Loadcase #1"
```

or

```
abaqus581 job=JobName input=InputDeck.inp interactive > status.sta
```

1.2 Adams

```
#!/bin/ksh
#

export LM_LICENSE_FILE=2003@serv123:/appl/soft/adams10.0/network/hpux10/license.dat

#  Submitting Adams job

/appl/soft/adams10.0/mdi -c ru-u i a101 inputdeck.acf ex 2
```

1.3 Ansys

1.3.1 Unix

```
ansys -V prod -a 5.6 -j ansystobi -m 100 -db 50 -r f < ansystobi.input\
> ansystobi.outputtext 2>&1
```

or

```
del mach3b.out
del mach.sye
ren OUTPUTFILE.DAT mach.sye
ansys57 -p ane3fl -j mach3 -i mach3b.inp -o mach3b.out
```

or

```
echo "Starting - Creating Element Centroid Data - ANSYS/AutoPWXC"
del mach3a.out
del POINTFILE.DAT
ansys57 -p ane3fl -j mach3 -i mach3a.inp -o mach3a.out
ren ecn.data POINTFILE.DAT
```

1.3.2 Windows

```
c:/ansys55/bin/intel/ansys55.exe -b ~~-p ANSYSUL~~ -I input.log -O contact.out"
```

1.4 Autoform

Set up the job in Autoform-GUI (xaf). Write input/control files. These files have the extensions:

mat material file

af geometry information

inp input deck

ctl control file for special control setup such as wrinkling

Then you can run Autoform from the command line

```
af_3.04 -jn jobname -bg
```

1.5 CAEFEM

```
c:/caefem40/wcaefem.exe  -neu Newfly.neu -an ls
```

1.6 CFX5 Solver

```
cfx5solve.exe -def ./static_mixer.def
```

CFX5 Preprocessor

```
cfx5build.exe -b -play input.jou
```

1.7 DADS

Sample Script.

```
#!/bin/csh
setenv CADSI /home/DADS/dads85/dadsaux/
setenv LM_LICENSE_FILE /home/DADS/dads85/dadsaux/license.dat
set path = (/home/DADS/dads85/execute_irix5_32 /usr/bin/X11 $path)
#
```

```
cd /home/anybody/DADS/dads85/bat
rm -f bat.fm2
dads -nomenu bat.def << EOF
save formatted bat
exit no
EOF
rundads 2 bat bat bat bat </dev/null

dadsgraph -nomenu << EOF
open bat
cu t ball:y
cu t ball:x
write curve stroke.dat
exit no
EOF
```

1.8 Excel

Excel execution is tightly integrated with iSIGHT. You need a specific version of the cscript.exe shell (5.1.0.4615) though. To find out the version on your system, navigate to `C:\WINNT\System32` using the Windows Explorer and right click on cscript.exe. Select Properties and the Version-tab.

If you have an old version, you can download updates from the MS-Webpage:

http://msdn.microsoft.com/scripting/default.htm?/scripting/windowshost/download/default.htm

For customers from german speaking countries it is necessary to set the decimal separator to a dot "." instead of a comma "," . You can do this in the System.Country Settings.Numbers menu of Windows.

In some cases however, you have to change the country settings for the user or even for the system default to an english speaking country in order to get the dot in Excel batch execution.

1.9 FEMAP

```
c:/caefem40/femap50/femap.exe newhub2.mod
```

1.10 Fluent

1.10.1 Unix

```
#!/bin/csh
setenv DISPLAY happy:0.0
cd /users/david/ugif
#local --- fluent 3d -i test.jou
#remote --- fluent 3d -i test.jou
```

```
/hosts/runner2/export/proj/apps/Fluent.Inc/bin/fluent 3d -i gmtube.jou
```

1.10.2 Windows

```
C:/FLUENT.INC/ntbin/fluent.bat -v3d -wait -r5.4 -i myfluentrunauto.jou
```

1.11 Gambit

```
gambit -in mygambit.jou
```

1.12 GT-Power

GT-power may have changed the input file format. Its extension is ”gtm”. It’s a binary file. The old file input file, ”.dat” is a text file. But you can transform ”gtm” file into ”.dat” file from GT-power GUI. In the GT-Power GUI is a feature called ”create .dat” file.

1.12.1 Unix

```
/usr/gti/bin/gtpowerx 5.1.0 BMW_E85_M54_009.dat dummy off off
```

1.12.2 Windows

```
GTpower.exe GTpowerX Input.dat
```

1.13 Hydsim

```
@echo off
set DIR=.
set ASTLM_LICENSE_FILE=%DIR%\bin\hydsim.lic
echo simulation unit_injector_isight > cmd.txt
%DIR%\bin\hydsim <cmd.txt >hydsim.out
del cmd.txt animationFilenames *.lst *.anm *.GAD unitinj_opt1.GID hydsim.out
```

DIR is the install dir of Hydsim

Note: AVL will change to FLEX*lm*-licensing in the future.

1.14 ICEM CFD

```
icemcfd -batch -script icem_script
```

1.15 IDEAS

```
ideas -q -t mm -p "iSIGHT" -m "/home/kevin/test.mf1" -e "/f pr r fil pasedinput.prg okay"
```

1.16 LSDYNA

```
setenv LSTC_FILE ../LSTC_FILE
```

```
ls940.2b.nov.22 i=your_input_file memory=10000000 > your_output_file
```

1.17 Matlab

```
matlab input.m
```

or

```
matlab -r input.m -logfile output.dat
```

To turn of the splash window at startup use:

```
matlab.exe /nosplash
```

on Windows or,

```
matlab -nosplash
```

on Unix.

Sometimes a -n option is needed to get Matlab running.

1.18 matrixX

```
xmath -tty < opti.ms > goober.output
```

1.19 Maze and dyna 2d

```
poly < splint
maze c=w.maz  i=maze.dat  o=w.d2i f=hsp
dyna2d w
./cf1
```

1.20 MoldFlow

```
#!/bin/csh
mfl -input moldflow.001
echo Fast Fill Analysis Completed!
```

1.21 Nastran

1.21.1 Unix

```
nastran d200g13-2 batch=no
```

1.21.2 Windows

This is an excerpt of an example description file. You can only use nastranw.exe in Windows2000 machine. Because of running in the background, you have to add time delay command in Tcl block. or check that the output file is created or not.

```
 SimCodeProcess nast
    Program: "C:/msc/bin/nastranw.exe"
    ElapseTime: 5m
    Prologue
        WriteInputSpecs: generate1
    Epilogue
        Tcl
            after 3000
            set OnBuild [ file exists leafspring.pch ]
            puts " Nastran start... "
            while { $OnBuild == 1 } {
               after 3000
               set OnBuild [ file exists leafspring.pch ]
               puts " Nastran running.... "
            }
        End Tcl
        ReadOutputSpecs: leafspringout
    Stdin: generate1
    Execution: "$Program leafspring.bdf scr=yes bat=no <$Stdin"
End SimCodeProcess nast
```

1.22 Ncode

```
atd /inp=RightRear /out=RightRear /ov=y /spa=n /head=0 /all=n /tcol=1 /dcol=2
asd /inp=RightRear /out=RightRear /ov=y /otype=psd
dta /inp=RightRear.psd /out=RightRear /ov=y /head=n /mul=n /del=s
```

1.23 PAMCRASH

```
  pamcrash inp=<your_input_file> [out=<your_output_file>] [numpro=<cpus>] [que=<batch_queue>]
```

input input file name. This parameter is required

output output file name. If omitted the output file will have the name of the input file but the suffix "out"

cpus number of cpus to be used by the job.

Use PamView to extract relevant parameters from the PAMCRASH output.

```
/soft/esi/pamcrash/bin/pamv -ns -c boxbeam.cmdf
```

1.24 Patran

You can run Patran in batch mode with GUI of Patran

```
patran -sfp flow2d_new.ses
```

or without GUI of Patran

```
patran -sfp -b flow2d_new.ses
```

where `flow2d_new.ses` is an input session file of Patran.

1.25 Pro/E

```
proe2001.bat  pro_wait  -g:no_graphics  trailfile.txt
proe2000i trailfile.txt
```

1.26 RADIOSS

```
  setenv RADIR /export/home/isis/RADIOSS
setenv LM_LICENSE_FILE $RADIR/license.dat

s41b_sgi64 < your_D00_file
e41b_sgi64 < your_D01_file
th++/th_v41b_SGI6 -b yourpostcommands.batch
```

1.27 SYSNOISE

script.

```
 /home3/sysnoise/5.4/bin/sysnoise -nogui  -if  sysnoise.inp > sysnoise.out
```

(You should change path and revision number.)

1.28 Tycon

```
REM @echo off
set DIR=.
set ASTLM_LICENSE_FILE=%DIR%\bin\tycon.lic
set TYCON_ERROR=%DIR%\bin\error
set AVLAST_HOME=%DIR%
type simulation | %DIR%\bin\tycondyn  >tycon.log
REM %DIR%\bin\tycondyn <simulation >tycon.log
rename fort.3 tycon.out
%DIR%\bin\tycondbs
del *.TMP *.RSF *.NID *.PRO *.STP *.ACC *.PAT *.BIF *.WRK *.PDA
```

Note: AVL will change licensing to FLEX*lm* in the future.

1.29 UniGraphics

```
@echo off

echo Starting Unigraphics - Create Geometry
set UGII_LICENSE_FILE=27000@walleye
set UGII_FLEX_BUNDLE=ADVDES
set UGII_FLEXLM=c:\ugs160\ugflexlm
set UGII_BASE_DIR=c:\ugs160
set UGII_ROOT_DIR=c:\ugs160\ugii
set path=%UGII_ROOT_DIR%;%path%
del mach3.prt
call c:\mlong\PWxsection\AutoPWXC\ug_update_expressions.exe -p subrotor_orig.prt -e mach3.exp
echo "done" >> ugbatch.out
echo "Finished Unigraphics - Create Geometry"

echo Finished - Creating Element Centroid Data - ANSYS/AutoPWXC
```

2 Generic Scripts

2.1 C-Shell

(Similar for Bourne Shell and K-Shell)

Example of passing input vairables and saving results to different directories for each iteration

```
#!/bin/csh -f

set currentdir = $1
set casename = $2
set rc = $3
set coupling = $4
set numsr = $5
set sr = ( $6 )
unalias cp
unalias rm
cd $currentdir
set i = 1
while ( $i <= $numsr )
   mkdir sr${sr[$i]}

   if ($i == 1) then
        cp -r ../../ufile ufile
        ln -s ${casename}.parameter parameter
starlink<<EOF
${casename}.exe
s
y
n
EOF
   endif

${casename}.exe >& starcd.log <<EOF
$casename
$coupling,0,$sr[$i]
EOF

   if ($i == 1) then
        ../../restart.script $casename $currentdir
   endif

   mv ${casename}.summary sr{$sr[$i]}/output.summary
   mv ${casename}.*.dat sr{$sr[$i]}/.
   mv starcd.log sr{$sr[$i]}/.
   cp ${casename}.pst sr{$sr[$i]}/.
   @ i++
end
```

2.2 ksh Script

When running in parallel from an NFS mounted directory, input and output can become the bottleneck if the simcodes write out large amounts of data.

This script runs an Abaqus and Nastran calculation in a local directory to avoid this.

Abaqus had problems to converge for some input parameters, so the Abaqus-status is stated explicitely in the exit command.

```
#!/bin/ksh
export PATH=$PATH:/ford/thishost/unix/bin:/apps/bin
```

```
UNIQUE=$((ps -a&& top -d2 && date)| sum | cut "-d " -f1)
#echo $UNIQUE

WORKDIR=$(pwd)

mkdir /tmp/bothrun_$UNIQUE
cp  nastran_base.dat abaqus_mod.inp /tmp/bothrun_$UNIQUE

cd /tmp/bothrun_$UNIQUE
mkdir /tmp/bothrun_$UNIQUE/scratch

nastran nastran_base.dat batch=no sdir=./scratch
abaqus interactive j=abaqus_mod

grep -q "THE ANALYSIS HAS COMPLETED SUCCESSFULLY" abaqus_mod.sta
# $? has the exit status of the last executed command. It is 0 if
# grep found the given string in Abaqus output.
ABAQUS_STATUS=$?

cp abaqus_mod.sta nastran_base.f06 $WORKDIR
cd -
rm -rf /tmp/bothrun_$UNIQUE
exit $ABAQUS_STATUS
```

2.3 Perl

Generic Perl script to execute a simcode and check the output to single a completion. Works well with commands that spawn jobs. This will play around with waiting for a file to exist.

```
$file = "phoo.txt";
$rc = system("perl foo2.pl")
  or die "unable to run foo2: $!\n";
print "ran foo2 rc=$rc msg=$!\n";
$size = 7;
$n = 0;
MainLoop:
while (1) {
  $n ++;
  sleep(2);
  @s = stat($file)
    or die "cannot stat $file: $!\n";
  next if $s[$size] < 10;
  open(A, "< $file")
    or die "cannot open file: $!\n";
  while (defined($a = <A>)) {
    last MainLoop if $a =~ /last line/;
  }
  close(A);
}
close(A);
chomp($a);
print "Done!  loops=$n  size=$s[$size]  line=$a\n";
```

2.4 Remote Shell

If you have to use a remote shell from within iSIGHT, don't forget to give the -n option to the rsh command

```
rsh hostname -n scriptname
```

otherwise the process will hang.

And remeber that in HP-UX (at least under ksh) the name for rsh is remsh. rsh there will give you a restricted shell, which is certainly not what you want.