Running (remote) jobs

To start a job, the *ecflow_server* uses the content of the ECF_JOB_CMD *variable*.
By modifying this variable, it is possible to control **where** and **how** a *job file* will run.
The command should be used in conjunction with the *variable* ECF_JOB and ECF_JOBOUT.
The ECF_JOB variable contains the *job file* path and ECF_JOBOUT contains
the path of a file where the output of the job will be written. The default command:

> ⚠ ECF_JOB_CMD = %ECF_JOB% 1> %ECF_JOBOUT% 2>&1 &

Let us run the tasks on a remote machine. For that, we could use the UNIX command ssh.
We would like the name of the host to be defined by a *variable* called HOST.
We assume that all the files are visible on all the hosts, e.g. using NFS.

In the examples below replace the string ?????? with a host name of your choice.

> ⚠ The environment of a task running on a remote host can be different from that of a task running locally.
> It depends on how your system is set up. The *head.h* should already be using the correct PATH, to allow *child command*s to be used.
> If not add the following line into your *head.h* file before the call to *ecflow_client* –init
>
> > export PATH=$PATH:/usr/local/apps/ecflow/%ECF_VERSION%/bin

To use ssh requires your public key to be available on the destination machine.
Check if you can log on to the remote machine through ssh without a password check.
If you need to enter a password you will need to add your public key on the destination machine. To do this issue the following commands:

no password for ssh connection

```
REMOTE_HOST=??????  # change me
ssh $USER@$REMOTE_HOST mkdir -p \$HOME/.ssh      # if you are prompted for a password use your Training
password that was provided
cat $HOME/.ssh/id_rsa.pub || ssh-keygen -t rsa -b 2048
cat $HOME/.ssh/id_rsa.pub | ssh $USER@$REMOTE_HOST 'cat >> $HOME/.ssh/authorized_keys'
```

Modify the *family* f5 so that all its tasks will run on another machine in the classroom.

Text

```
# Definition of the suite test
suite test
 edit ECF_INCLUDE "$HOME/course"
 edit ECF_HOME    "$HOME/course"
 limit l1 2

 family f5
     edit HOST ??????
     edit ECF_OUT /tmp/$USER
     edit ECF_JOB_CMD "ssh %HOST% 'mkdir -p %ECF_OUT%/%SUITE%/%FAMILY% && %ECF_JOB% > %ECF_JOBOUT% 2>&1 &'"
     inlimit l1
     edit SLEEP 20
     task t1
     task t2
     task t3
     task t4
     task t5
     task t6
     task t7
     task t8
     task t9
 endfamily
endsuite
```

If your login shell is csh, you should define ECF_JOB_CMD as:

```
edit ECF_JOB_CMD "ssh %HOST% 'mkdir -p %ECF_OUT%/%SUITE%/%FAMILY%; %ECF_JOB% >& %ECF_JOBOUT%'"
```

Python

In python modify the function create_family_f5() created in the earlier page, to add HOST,ECF_OUT,ECF_LOGHOST,ECF_LOGPORT, and ECF_JOB_CMD:

$HOME/course/test.py

```
import os
from ecflow import Defs,Suite,Family,Task,Edit,Trigger,Complete,Event,Meter,Time,Day,Date,Label, \
                   RepeatString,RepeatInteger,RepeatDate,InLimit,Limit

def create_family_f5() :
    return Family("f5",
            InLimit("l1"),
            Edit(SLEEP=20,
                 HOST='?????',
                 ECF_OUT = '/tmp/%s' % os.getenv("USER"),
                 ECF_LOGHOST='%HOST%',
                 ECF_LOGPORT='?????',  # port=$((35000 + $(id -u))) run this on the command line
                 ECF_JOB_CMD="ssh %HOST% 'mkdir -p %ECF_OUT%/%SUITE%/%FAMILY%; %ECF_JOB% > %ECF_JOBOUT% 2>&1
&'"),
            [ Task('t{}'.format(i)) for i in range(1,10) ] )

print("Creating suite definition")
home = os.path.join(os.getenv("HOME"),"course")
defs = Defs(
        Suite("test",
            Edit(ECF_INCLUDE=home,ECF_HOME=home),
            Limit("l1",2),
            create_family_f5()))
print(defs)

print("Checking job creation: .ecf -> .job0")
print(defs.check_job_creation())

print("Checking trigger expressions")
assert len(defs.check()) == 0,defs.check()

print("Saving definition to file 'test.def'")
defs.save_as_defs("test.def")
```

Logserver

We can view the output on the remote machine (class??) by using a log server.

This assumes you have defined variables ECF_LOGHOST and ECF_LOGPORT in your definition.

Launch the log server on a remote machine:

```
ssh $USER@class01 /usr/local/apps/ecflow/5.5.1/bin/ecflow_logserver.sh -d /tmp/$USER -m /tmp/$USER:/tmp/$USER
```

What to do

1. Modify PATH environment variable in head.h
2. Change the *suite definition*
3. Replace the *suite definition*
4. It may not work immediately. Have a look in the file $HOME/course/*host.port*.ecf.log to see why.
5. Add a uname -n to your ECF script to see what machine the task is running on.
6. What do you need to do in order to have the task **/test/f5/t9** run on another machine? Try your solution.
7. Create a log server, to access the remote output