

Instructions for running on DoD HPC computers

01. Websites for access, account control and documentation

Getting Started to access HPC supercomputers:

<https://centers.hpc.mil/users/index.html>

Documentation on HPC supercomputers

<https://centers.hpc.mil/systems/unclassified.html>

piE Account control:

<https://ieapp.erdc.hpc.mil>

Kerberos validation:

<https://ieapp.erdc.hpc.mil/info/kerberosValidate.jsp>

For Navy Web Terminal Portal link:

<https://portal.navydsrc.hpc.mil/portal/home/>

For Web session portal:

<https://portal.navydsrc.hpc.mil/portal/group/home/apps>

Kerberos Yubikey login:

<https://openid.hpcmp.hpc.mil/login>

<https://openid.hpcmp.hpc.mil/login?id=krb5>

02. Web Portal Access

Access computers from "Web Terminal" on Navy Portal Web Link once logged into the "Apps" page. Once in the webterminal window, you will have a user login prompt similar to "username@vappsg-0003".

Your home directory will be "/u/home/username".

Type the commands:

a. "kshell"

b. "kinit username@HPCMP.HPC.MIL" or just "kinit"

c. Enter HPC Password when prompted.

d. Enter Yubikey code when requested.

03. Logging in to supercomputers

From here you can login to your accessible supercomputers:

ssh username@copper.ors.hpc.mil

ssh username@haise.navo.hpc.mil

ssh username@conrad.navydsrc.hpc.mil

ssh username@gordon.navydsrc.hpc.mil

04. General RAMS tech info:

RAMS is compiled within "bin.rams". The "Makefile" references "include.mk" at the top of the file. The "include.mk" file contains environment paths and compile commands. Make sure to edit the file for your compilers and paths.

05. Running Batch mode scripts on listed computers below:

a. Write up a simple batch script in your default Linux shell.

b. Make sure your script does not have any CR/LF trailing characters that might not be compatible with Linux. To eliminate trailing CR/LF for Linux, run:

sed -i -e 's/\r\$//' scriptnamehere

c. To run batch script job type "**qsub scriptnamehere**".

d. To check status of jobs type "**qstat -u username**".

e. For interactive batch session type something like:

qsub -A projectnumber -l select=2:ncpus=32:mpiprocs=32

-q background -l walltime=000:05:00 -I

And then execute command from \$WORKDIR:

aprun -n 64 ./rams-6.1.22_dm -f ./RAMSIN 1>z.out 2>&1 &

06. Specific settings and commands for various supercomputers

"Copper":

- a. \$HOME is /u/username
- b. \$WORKDIR is /work/username
- c. \$CENTER is /p/cwfs/username
- d. Note that we have had no problem using our own version of parallel HDF5 located in "misc/ hdf5-1.8.9-64bit-parallel" within the main RAMS directory.
- e. To compile RAMS in MPI/PARALLEL mode to run on compute nodes, use compile file "include.mk.copper" renamed to "include.mk" and set "PAR_DEFS=-DRAMS_MPI". Use module "craype-interlagos" (which is default) rather than module "craype-target-native". Do "module swap craype-target-native craype-interlagos".
- f. To compile RAMS in SEQUENTIAL mode on the login node, do: "module swap craype-interlagos craype-target-native" and set "PAR_DEFS=" in the include.mk file you are using. This turns off use of MPI and allows you to run simple jobs on the login node.
- g. PGI modules for compiling are setup by default. If you want to use Intel compilers, do "module swap PrgEnv-pgi PrgEnv-intel". The Intel compiler seems to be working best on copper.
- h. Setting for the include.mk files:

```
-----
#For compilers
F_COMP=ftn
F_OPTS=-free -O1 -fp-model strict  #For INTEL
F_OPTS=-Mfree -O2 -Kieee           #For PGI
LOADER=$(F_COMP)
LOADER_OPTS=$(F_OPTS)
C_COMP=cc
C_OPTS=-O3 -DUNDERScore -DLITTLE
-----
```

- i. A simple PBS run script might look like the following, where "ncpus=32" for **copper** is non-changeable.

```
-----
#!/bin/bash
#PBS -A projectnumberhere
#PBS -l select=2:ncpus=32:mpiprocs=32
#PBS -l walltime=000:05:00
#PBS -q standard
#PBS -N z.out
#PBS -j oe
cd ${WORKDIR}
aprun -n 64 ./rams-6.1.22_dm -f ./RAMSIN
-----
```

"Haise":

- a. \$HOME is /u/home/username
- b. \$WORKDIR is /scr/username
- c. \$CENTER is /p/cwfs/username
- d. Note that we have had no problem using our own version of parallel HDF5 located in "misc/ hdf5-1.8.9-64bit-parallel" within the main RAMS directory.
- e. To compile RAMS in MPI/PARALLEL mode to run on compute nodes, use compile file "include.mk.haise" renamed to "include.mk" and set "PAR_DEFS=-DRAMS_MPI".
- f. Intel compiler appears setup for the default. This compiler appears to work fine for RAMS test runs. You can swap to PGI if desired but I have not tested using PGI on "haise".

- g. Changing between running parallel or sequential jobs does not appear to require swapping modules or recompiling. For compiling with IFORT, using "mpiifort" and "mpiicc" appears to work for running sequentially or parallel. Sequential mode is useful for simple MAKESFC or MAKEVFILE run on the login node.

h. Setting for the include.mk files:

```
-----
#For default INTEL compiler
F_COMP=mpiifort
F_OPTS=-free -O1 -fp-model strict
LOADER=$(F_COMP)
LOADER_OPTS=$(F_OPTS)
C_COMP=mpiicc
C_OPTS=-O3 -DUNDERScore -DLITTLE
-----
```

- i. A simple PBS run script might look like the following, where "ncpus=16" for **haise** is non-changeable.

```
-----
#!/bin/bash
#PBS -A projectnumberhere
#PBS -l select=32:ncpus=16:mpiprocs=16
#PBS -l walltime=000:05:00
#PBS -q standard
#PBS -N z.out
#PBS -j oe
cd ${WORKDIR}
mpirun -n 512 ./rams-6.1.22_dm -f ./RAMSIN
-----
```

"Conrad":

- \$HOME is /p/home/username
- \$WORKDIR is /p/work1/username
- \$CENTER is /p/cwfs/username
- Note that we have had no problem using our own version of parallel HDF5 located in "misc/ hdf5-1.8.9-64bit-parallel" within the main RAMS directory.
- To compile RAMS in MPI/PARALLEL mode to run on compute nodes, use compile file "include.mk.conrad" renamed to "include.mk" and set "PAR_DEFS=-DRAMS_MPI". Use module "craype-haswell" (which is default) rather than module "craype-target-native". Do "module swap craype-target-native craype-haswell".
- To compile RAMS in SEQUENTIAL mode on the login node, do: "module swap craype-haswell craype-target-native" and set "PAR_DEFS=" in the include.mk file you are using. This turns off use of MPI and allows you to run simple jobs on the login node.
- Cray compiler modules for compiling are setup by default. If you want to use Intel compilers, do: "module swap PrgEnv-cray PrgEnv-intel".
- Setting for the include.mk files:

```
-----
#For compilers
F_COMP=ftn
F_OPTS=-free -O1 -fp-model strict #For INTEL
F_OPTS=-Mfree -O2 -Kieee #For PGI
LOADER=$(F_COMP)
LOADER_OPTS=$(F_OPTS)
C_COMP=cc
C_OPTS=-O3 -DUNDERScore -DLITTLE
-----
```

-
- i. A simple PBS run script might look like the following, where
"ncpus=32" for **conrad** is non-changeable.
-

```
#!/bin/bash
#PBS -A projectnumberhere
#PBS -l select=2:ncpus=32:mpiprocs=32
#PBS -l walltime=000:05:00
#PBS -q standard
#PBS -N z.out
#PBS -j oe
cd ${WORKDIR}
aprun -n 64 ./rams-6.1.22_dm -f ./RAMSIN
-----
```

"Gordon":

- a. \$HOME is /p/home/username
- b. \$WORKDIR is /p/work1/username
- c. \$CENTER is /p/cwfs/username
- d. Note that we have had no problem using our own version of parallel HDF5 located in "misc/ hdf5-1.8.9-64bit-parallel" within the main RAMS directory.
- e. To compile RAMS in MPI/PARALLEL mode to run on compute nodes, use compile file "include.mk.gordon" renamed to "include.mk" and set "PAR_DEFS=-DRAMS_MPI". Use module "craype-haswell" (which is default) rather than module "craype-target-native". Do "module swap craype-target-native craype-haswell".
- f. To compile RAMS in SEQUENTIAL mode on the login node, do: "module swap craype-haswell craype-target-native" and set "PAR_DEFS=" in the include.mk file you are using. This turns off use of MPI and allows you to run simple jobs on the login node.
- g. Cray compiler modules for compiling are setup by default. If you want to use Intel compilers, do: "module swap PrgEnv-cray PrgEnv-intel".
- h. Setting for the include.mk files:

```
#For compilers
F_COMP=ftn
F_OPTS=-free -O1 -fp-model strict #For INTEL
F_OPTS=-Mfree -O2 -Kieee #For PGI
LOADER=$(F_COMP)
LOADER_OPTS=$(F_OPTS)
C_COMP=cc
C_OPTS=-O3 -DUNDERScore -DLITTLE
-----
```

- i. A simple run PBS script might look like the following, where
"ncpus=32" for **gordon** is non-changeable.
-

```
#!/bin/bash
#PBS -A projectnumberhere
#PBS -l select=2:ncpus=32:mpiprocs=32
#PBS -l walltime=000:05:00
#PBS -q standard
#PBS -N z.out
#PBS -j oe
cd ${WORKDIR}
aprun -n 64 ./rams-6.1.22_dm -f ./RAMSIN
-----
```