The *reservable front end* feature allows you to reserve dedicated access to an Electra compute node for up to 90 days. You can use the node the same way you would use a PFE—to submit jobs and perform other tasks such as post-processing, visualization, and so on—except that the node will be reserved specifically for you (or your group) so you can use all of the node's processor and memory resources. No other users will have access to the node.

## Reserving a Dedicated Node

To request a dedicated node, run the "reservable front end" command (pbs_rfe), specifying what type of node you want and for how long. The command makes a PBS advanced reservation for a node, and tells you which node is reserved.

You can submit batch or interactive jobs, check status, and so on, from your reserved node just as you would from a PFE.

## pbs_rfe Command Usage

Run pbs_rfe --help to see the command options and arguments, as shown below:

```
$ pbs_rfe --help
usage: pbs_rfe [-h] [--debug] [--duration DURATION] [--group GROUPS]
               [--model MODEL] [--name NAME] [--starttime STARTTIME]
               [--user USERS] [--version] [-W BILL_GROUP]
               [{request,delete,status,which}] [clue]

...

positional arguments:
  {request,delete,status,which}
                        action to perform
  clue                  node, reservation id, or reservation name

optional arguments:
  -h, --help            show this help message and exit
  --debug, -d           increase debugging verbosity
  --duration DURATION, -D DURATION
                        how long to reserve front-end. Format: [days+]hours
  --group GROUPS, -G GROUPS
                        groups whose members can also use front-end
  --model MODEL, -m MODEL
                        processor type
  --name NAME, -N NAME  name to give reservation
  --starttime STARTTIME, -R STARTTIME
                        when reservation should start. Format as for date(1)
                        command
  --user USERS, -U USERS
                        other userid[s] who can also use front-end
  --version             show program's version number and exit
  -W BILL_GROUP, --group_list BILL_GROUP
                        GID to charge reservation to

In addition to the default "request" action, you can "delete" a
reservation, get the "status" of a reservation, or see "which" node is
assigned to a reservation.

 pbs_rfe delete resv_id       cancel a reservation (only one at a time)
 pbs_rfe status               list your reservations and their statuses
 pbs_rfe which                display the nodes assigned to you
```

The --starttime option allows you to specify when the reservation will start (the default is to start as soon as possible).

The --user option lets you specify other users who will be permitted to log in to your reserved host with you. For example, this might be useful during a shared debugging session. You can also use the --group option to permit all users whose primary group is in the specified list.

If you forget which node is assigned, you can run pbs_rfe which to identify the correct node. For example:

```
$ pbs_rfe which
r617i0n13
```

You can also use the which option in scripts—for example, when you need to learn which node is yours.

To find out the status of your reservation, run pbs_rfe status. For example:

```
$ pbs_rfe status
Resv name    Resv ID   User      ST        Start / End          Node
-----------  --------  --------  --  ------------ / ------------  ----
FE_zsmith    R3286463  zsmith    CO  Feb 05 15:38 / Feb 15 15:38  r617i0n13
```

Note: You cannot use the node until the reservation starts. PBS will send you an email at the start of the reservation period.

When the reservation is ready, the status (ST) will change from CO to RN.

## Ending the Node Reservation

When you are finished with a reserved node, you should delete the reservation using the pbs_rfe delete command. For example:

```
$ pbs_rfe status
Resv name    Resv ID   User      ST        Start / End          Node
-----------  --------  --------  --  ------------ / ------------  ----
FE_zsmith    R3286638  zsmith    CO  Feb 09 10:00 / Feb 10 10:00  r601i2n6

$ pbs_rfe delete R3286638
R3286638.pbspl1.nas.nasa.gov deleted.
```

## Setting Up the Node

While you can simply SSH into the reserved node from a PFE to run commands, the node is more useful if you start a screen or VNC session there. This allows your sessions on the node to survive even if the PFE you logged into crashes.

Note: This section describes how to set up VNC on the reserved node. For information on using the screen tool, see **man screen** or **info screen**.

## VNC Setup

If you want to use a graphical user interface (GUI) to run commands on your node, you'll want to use VNC to set that up. We recommend reading the article VNC: A Faster Alternative to X11 to learn more about VNC and how to establish a VNC session. However, using VNC on a reserved node is simpler than described in that article, once a few preliminary steps are accomplished. These steps are described below.

Before You Begin: Set up SSH passthrough.

Complete these steps:

1. If you do not have the **$HOME/.vnc/xstartup** file, create one on a PFE by following steps 1 and 2 in VNC: A Faster Alternative to X11.
2. Using ssh, log into your reserved node and start vncserver. Do not use the -localhost option.

   ```
   $ ssh r601i1n3 vncserver
       New 'r601i1n3:1 (zsmith)' desktop is r601i1n3:1
       Starting applications specified in /u/zsmith/.vnc/xstartup
       Log file is /u/zsmith/.vnc/r601i1n3:1.log
   ```

3. Create a network tunnel from your workstation to the reserved node. The following command logs you into a PFE with a tunnel that leads to the node:

   ```
   $ ssh -L 5901:node_name:5901 pfe
   ```

   where node_name is the name of the reserved node (r601i1n3, in our example). Note that since you are the only user on your node, the 5901 parts of the command line are constant, unlike when you run vncserver on a PFE node.
4. Start your VNC client on your workstation and connect to localhost:5901. You will be connected to your graphical environment on your reserved node.

## Limitations

- The compute node reserved via pbs_rfe does not have the PBS_NODEFILE environment variable set by default. If you plan to run any application that relies on having PBS_NODEFILE set, you will have to set it manually. For example, if you try to run an MPI job on the node without presetting PBS_NODEFILE, you will get the following error message:

  ```
  name_of_compute_node> mpiexec -np X a.out
  Not invoked from a known Work Load Manager:
  o For PBS   : PBS_NODEFILE is not set.
  o For SLURM : SLURM_JOB_ID is not set.
  mpiexec.real error: Aborting.
  ```

  The solution is to create a file, for example a file named "*hfile*", and put the name of your reserved node in it as many times as you want MPI ranks. Then, set the PBS_NODEFILE environment variable to the path to *hfile*.

  ```
  (for bash)
  export PBS_NODEFILE=/path/to/hfile

  (for csh)
  setenv PBS_NODEFILE /path/to/hfile
  ```

- Compute nodes do not have all of the software packages that are installed on the PFEs. You might need to load the current pkgsrc module to get access to some commands. See Using Software Packages in pkgsrc for more information.

- /tmp on a node is small and takes memory away from programs.
- The reserved nodes don't have direct network access to hosts outside Electra. Use your PFE login to transfer files, to outside hosts.
- The cron daemon does not run on reserved nodes. Again, use a PFE for that type of activity.
- While reservations are allowed for up to 90 days, a reserved node might need to be rebooted sooner than that (e.g., for system patches), or you might run the node out of memory. You'll still have the node reserved after the reboot, but anything you were doing will be lost. Please be sure to save your work frequently.
- Currently, each user is limited to only one node reserved at a time.

## Account Charging

Reserved nodes are charged Standard Billing Units (SBUs) at the same rate as PBS jobs, whether you are actively using the node or not. Reservations are billed daily. Be sure to delete your reservation when you no longer need it.

To reduce costs, a few users can share the same reservation. (See the `--user` and `--group` options above, and Sharing with Multiple Users.

The PBS batch scheduler supports a feature called shrink-to-fit (STF). This feature allows you to specify a range of acceptable wall times for a job, so that PBS can run the job sooner than it might otherwise. STF is particularly helpful when scheduling jobs before an upcoming dedicated time.

For example, suppose your typical job requires 5 days of wall time. If there are fewer than 5 days before the start of dedicated time, the job won't run until after dedicated time. However, if you know that your job can do enough useful work running for 3 days or longer, you can submit it in the following way:

qsub -l min_walltime=72:00:00,max_walltime=120:00:00 *job_script*

When PBS attempts to run your job, it will initially look for a time slot of 5 days. When no such time slot is found between now and the dedicated time, it will look for shorter and shorter time slots, down to the minimum wall time of 3 days.

If you have an existing job that is still queued, you can use the qalter command to add these min_walltime and max_walltime attributes:

qalter -l min_walltime=*hh:mm:ss*,max_walltime=*hh:mm:ss job_id*

You can also use the qalter command to change the wall time:

qalter -l walltime=*hh:mm:ss job_id*

Note: For jobs on Endeavour, be sure to include both the job sequence number and the PBS server name, pbspl4, in the *job_id* (for example, 2468.pbspl4).

If you have any questions or problems, please contact the NAS Control Room at (800) 331-8737, (650) 604-4444, or by email at support@nas.nasa.gov.

During job execution, it can be useful to find out the amount of time remaining for your PBS job. This allows you to decide if you want to gracefully dump restart files and exit before PBS kills the job.

If you have an MPI code, you can call MPI_WTIME and see if the elapsed wall time has exceeded some threshold in order to decide if the code should go into the shutdown phase.

For example:

```
include "mpif.h"

real (kind=8) :: begin_time, end_time

begin_time=MPI_WTIME()
do work
end_time = MPI_WTIME()

if (end_time - begin_time > XXXXX) then
   go to shutdown
endif
```

In addition, the following library has been made available on NAS compute systems for the same purpose:

```
/u/scicon/tools/lib/pbs_time_left.a
```

To use this library in your Fortran code, you need to:

1. Modify your Fortran code to define an external subroutine and an integer*8 variable

   ```
   external pbs_time_left
   integer*8 seconds_left
   ```

2. Call the subroutine in the relevant code segment where you want the check to be performed

   ```
   call pbs_time_left(seconds_left)
   print*,"Seconds remaining in PBS job:",seconds_left
   ```

   Note: The return value from pbs_time_left is only accurate to within a minute or two.

3. Compile your modified code and link with the above library using, for example:

   ```
   LDFLAGS=/u/scicon/tools/lib/pbs_time_left.a
   ```

Many jobs running across multiple nodes perform setup and pre-processing tasks on the first node, then launch computations across many nodes, and finish with post-processing and cleanup steps running only on the first node. The post-processing and cleanup steps may take minutes or even hours, during which the remaining nodes in the job are idle. You can use the pbs_release_nodes command to release these idle nodes so that you are not charged for them, and they can be made available to other jobs.

Some users have already taken steps to minimize the number of idle nodes by separating their workflow into multiple jobs of appropriate sizes. The downside to this approach is additional time waiting in the queue for these multiple jobs. Using pbs_release_nodes may allow these jobs to be combined, reducing the amount of wait time in the queue.

## Node Tracking and Job Accounting

When you use the pbs_release_nodes command, the released nodes are removed from the contents of $PBS_NODEFILE. The job accounting system tracks when the nodes are released, and adjusts the amount of SBUs charged accordingly.

## Running the pbs_release_nodes Command

You can either include the pbs_release_nodes command in your job script or you can run it interactively, for example, on a PFE. If you run it in your job script, you can use the $PBS_JOBID environment variable to supply the *job_id*. You can use pbs_release_nodes multiple times in a single job.

There are two ways to use the pbs_release_nodes command. In both cases, the first node is retained.

- **Keep the first node and release all other nodes**
  If you want to retain the first node only, run the command as follows to release the remaining nodes:

  pfe21% pbs_release_nodes -j *job_id* -a

- **Release one or more specific nodes**
  If you want to retain one or more nodes in addition to the first node, run the command as follows to release specific idle nodes:

  pfe21% pbs_release_nodes -j *job_id node1 node2 ... nodeN*

To find out which nodes are being used by a job before you run the pbs_release_nodes command, use one of the following methods:

- **For interactive jobs:** Run qstat -f *job_id*. In the output, look for the nodes listed under the exec_vnode job attribute.
- **For job scripts:** Run cat $PBS_NODEFILE. The output provides a list of the nodes in use.

Note: $PBS_NODEFILE may have duplicate entries for nodes, depending on how resources were requested. See **man pbs_resources** for details.

## Sample Job Script with pbs_release_nodes Command

The following sample job script keeps the first node and releases all other nodes before running post-processing steps:

```
#PBS -S /bin/csh
#PBS -N cfd
#PBS -l select=4:ncpus=8:mpiprocs=8:model=sky_ele
#PBS -l walltime=4:00:00
#PBS -j oe
#PBS -W group_list=a0801
#PBS -m e
module load comp-intel/
module load mpi-hpe/mpt
cd $PBS_O_WORKDIR
mpiexec dplace -s1 -c 16-23 ./grinder < run_input > intermediate_output
# Before executing some post-processing steps that will run only on the
# first node of the job you can release all other nodes
pbs_release_nodes -j $PBS_JOBID -a
# Now run post-processing steps
./post_process < intermediate_output > final_output
# -end of script-
```