EPW School 2026 Cheat Sheet

TACC login, installed executables, Frontera/VISTA environment setup, GPU commands, troubleshooting, and file downloads

1. Login to TACC systems and PDF visualization

Use X11 forwarding when logging in if you need graphical output or PDF visualization.

Frontera login

```
ssh <username>@frontera.tacc.utexas.edu -Y
```

VISTA login

```
ssh <username>@vista.tacc.utexas.edu -Y
```

Opening a PDF on Frontera

Use evince to visualize PDF files:

```
evince <pdffilename>.pdf
```

If evince does not work, copy the PDF to your local machine using scp from a new local terminal, not from inside the SSH session:

```
scp <username>@frontera.tacc.utexas.edu:/full/path/to/<pdffilename>.pdf .
# For VISTA, use:
scp <username>@vista.tacc.utexas.edu:/full/path/to/<pdffilename>.pdf .
```

2. Installed executables and environments

Use these shared locations during the school. For most calculations, add the appropriate code directory to PATH or call executables from the corresponding bin directory.

Item	Path	Use
EPW v6.1s	/work2/05193/sabyadk/shared/EPW_6.1s/q-e	Main EPW v6.1s installation.
Quantum ESPRESSO 7.6 develop branch with EPW 6.1	/work2/05193/sabyadk/shared/QE_7.6dev/q-e	QE develop installation including EPW 6.1.
Quantum ESPRESSO 7.6 develop branch for Frontera GPU	/work2/05193/sabyadk/shared/QE_7.6devGPU/q-e	GPU-enabled QE develop installation on Frontera.
Wannier90 v3.1.0	/work2/05193/sabyadk/shared/W90/wannier90-3.1.0	Wannier90 installation for tutorials requiring Wannierization.
NVHPC compiler on Frontera	/scratch3/05193/sabyadk/nvidia/hpc_sdk_24.7	Needed only if building GPU-enabled QE on Frontera.
Virtual environment for EPWpy + EPW v6.1s	/work2/05193/sabyadk/shared/Virtual_env/venv/bin	Only for the last EPWpy tutorial on VISTA.

3. Running EPW v6.1s

EPW v6.1s was compiled with HDF5, so load the HDF5 module before running.

```
module load hdf5/1.10.4
export PATH=/work2/05193/sabyadk/shared/EPW_6.1s/q-e/bin:$PATH
```

Example launch pattern:

```
mpirun -np 1 pw.x -in scf.in > scf.out
```

4. Running QE v7.6devGPU on Frontera

Start an RTX development session before running GPU-enabled QE:

```
idev -N 1 -n 4 -p rtx-dev
```

Then load the GPU build environment:

```
module purge
deactivate
source /work2/05193/sabyadk/shared/QE_7.6devGPU/clean.sh
export PATH=/work2/05193/sabyadk/shared/QE_7.6devGPU/q-e/bin:$PATH
```

The clean.sh script contains the required NVHPC environment and prints the mpirun it will use:

```
/scratch3/05193/sabyadk/nvidia/hpc_sdk_24.7/Linux_x86_64/24.7/comm_libs/12.5/hpcx/hpcx-2.19/ompi/bin/mpirun
```

Launch calculations in the same manner as the non-GPU case, but disable CPU binding:

```
mpirun -np 1 --bind-to none pw.x -in scf.in > scf.out
```

5. Building GPU-enabled QE on Frontera, if needed

For the QE GPU tutorial, the recommended approach is to source the provided environment file instead of manually reconstructing the NVHPC environment.

```
source /work2/05193/sabyadk/shared/QE_7.6devGPU/clean.sh
```

Configure QE with CUDA support as follows:

```
./configure --enable-parallel --enable-openmp \
  --with-cuda=/scratch3/05193/sabyadk/nvidia/hpc_sdk_24.7/Linux_x86_64/24.7/cuda/12.5 \
  --with-cuda-cc=75 \
  --with-cuda-runtime=12.5 \
  --with-cuda-mpi=no
```

6. EPWpy environment on VISTA

For the last tutorial on VISTA, use the EPWpy + EPW v6.1s virtual environment:

```
source /work2/05193/sabyadk/shared/Virtual_env/venv/bin/activate
```

7. Troubleshooting

7.1 Frontera

- If libhdf is not found, load the HDF5 module:

```
module load hdf5/1.10.4
```

- If the wall time is more than about 2-3 times the CPU time without OpenMP, the run may be slowed by I/O. Set Lustre striping in the working directory before launching the calculation:

```
lfs setstripe -c 8 $PWD
```

7.2 VISTA

- If libpython.so is not found, or if there is a problem with Python execution, the GCC module is most likely not loaded. Load GCC as follows:

```
module load gcc/14.2.0
```

- For similar MPI-IO slowdowns on VISTA, use the following settings:

```
export OMPI_MCA_fs=ufs
export OMPI_MCA_fs_ufs_lock_algorithm=1
export OMPI_MCA_io=^ompio
export ROMIO_FSTYPE_FORCE="ufs:$SCRATCH"
# OR, for a specific working folder:
export ROMIO_FSTYPE_FORCE="ufs:/path_to_working_folder_where_MPIIO_occurs/"
```

8. Downloading tutorial files from Google Drive

Tutorial files can be downloaded directly from Google Drive using wget.

```
FILEID="1b8Mhsoc58sxeRZiuVIOh5qwvtG5so0FP"
FILENAME="Mon.4.Giannozzi.tar"

wget --no-check-certificate \
  "https://drive.google.com/uc?export=download&id=${FILEID}" \
  -O "${FILENAME}"
```

For example, for the QE tutorial on Monday:

```
https://drive.google.com/file/d/1b8Mhsoc58sxeRZiuVIOh5qwvtG5so0FP/view?usp=drive_link
```

For this link, the file ID is the last portion: 1b8Mhsoc58sxeRZiuVIOh5qwvtG5so0FP

9. Terminal basics and shell-script commands

Useful commands for navigating folders, checking environments, and running shell scripts during the tutorials.

Task	Command	Notes
Show current folder	`pwd`	Prints the present working directory.
List files	`ls -lh` `ls -la`	Human-readable sizes; include hidden files.

Change folders	`cd /path/to/folder`
	`cd ..`
	`cd -`
		Go to a folder; go up one level; return to previous folder.

Create a folder	`mkdir -p run1`	Creates the directory if it does not already exist.

Copy, move, remove	`cp input.in input.bak`
	`mv oldname newname`
	`rm filename`
		Use rm carefully; deleted files may not be recoverable.

Check file size and disk usage	`du -sh .` `ls -lh filename`	Useful before copying large tutorial folders.
Make a shell script executable	`chmod +x run.sh`	Shell scripts are not compiled; chmod gives execute permission.

Run a shell script	`./run.sh`
	`# or`
	`bash run.sh`
		./run.sh requires execute permission; bash run.sh does not.

Check which executable is used	`which pw.x`
	`which ph.x`
	`which epw.x`
	`echo $PATH`
		Useful when multiple QE/EPW versions are available.

Inspect output while a job runs	`tail -f scf.out`
	`less scf.out`
	`grep -i "error" scf.out`
		Follow a running output; browse a file; search for errors.

Module commands	`module list`
	`module avail hdf5`
	`module load hdf5/1.10.4`
	`module purge`
		Check, search, load, or reset modules.

Check running processes	`ps -u $USER` `jobs`	Shows your processes and shell background jobs.

Tip: run scp from your local computer terminal when copying files from Frontera or VISTA to your laptop.

10. Slurm batch jobs and interactive dev sessions

Use this section as a starting template for running EPW/QE calculations on Frontera. Replace the job name, reservation, queue, executable, and input file as needed for each tutorial.

10.1 Batch job submission with sbatch

Create a file such as job.ph.sh with the following content. The reservation name changes with the date/tutorial, and the queue should be normal for Frontera CPU jobs or rtx for Frontera GPU jobs.

```
#!/bin/bash
#SBATCH -J job.ph                          # Job name
#SBATCH -N 1                               # Total number of nodes
#SBATCH --ntasks-per-node 48               # MPI tasks per node
#SBATCH -t 00:30:00                        # Run time (hh:mm:ss)
#SBATCH -A DMR23048                        # Allocation/project
#SBATCH --reservation=MATCSSI_Norm_June15 # Reservation; changes with tutorial/date
#SBATCH -p normal                          # Queue: normal for CPU, rtx for GPU

# Launch MPI code
module load hdf5/1.10.4
export OMP_NUM_THREADS=1
export PATHQE=/work2/05193/sabyadk/shared/EPW_6.1s/q-e

mpirun -np 48 ${PATHQE}/bin/ph.x -in ph.in > ph.out
```

Submit job using:

```
sbatch job.ph.sh
```

Common Slurm commands:

Command	Purpose
sbatch job.ph.sh	Submit the batch job.
squeue -u $USER	Check your queued/running jobs.
scancel <jobid>	Cancel a submitted job.
tail -f slurm-<jobid>.out	Follow the Slurm output file while the job runs.

10.2 Interactive development session with idev

For an interactive development session, use idev with the allocation, node count, number of MPI tasks, queue, and reservation supplied for the tutorial.

```
idev -A DMR23048 -N 1 -n <number_of_tasks> -p <queue> -r <reservation>
```

Typical choices are:

Use case	Queue	Example
Frontera CPU	normal	idev -A DMR23048 -N 1 -n 48 -p normal -r MATCSSI_Norm_June15
Frontera GPU	rtx	idev -A DMR23048 -N 1 -n 4 -p rtx -r <reservation>

After the interactive session starts, load the appropriate environment and launch the calculation with mpirun, just as in the batch script.