Available Software

NYU HPC provides a range of third party and open source software for general use. To see the list of available software on a cluster, please use the Modules command. The resulting output lists the name of the software/compiler used to create the binary and version number:

```
$ module avail -t
```

The software list on this page is no longer current - please type "module avail" on the cluster for the current list, and see Accessing software via Environment Modules for more information
Available Software on Clusters

Select the Cluster USQ Bowery Cardiac

Note
Always, run the command `module avail` from the terminal to view the up to date available software.

General List

This page is retained from an earlier version of the HPC wiki only for reference.

Compilers

There are several Fortran, C, and C++ compilers available on our systems.

<table>
<thead>
<tr>
<th>Compiler Type</th>
<th>Name</th>
<th>Source</th>
<th>Documentation</th>
<th>Cluster availability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>C Compilers</strong></td>
<td>c99</td>
<td>Standard C</td>
<td>man c99</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>g9e</td>
<td>GNU C</td>
<td>man g9e</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>icc</td>
<td>Intel C</td>
<td>man icc</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td><strong>C++ Compilers</strong></td>
<td>g++</td>
<td>GNU C++</td>
<td>man g++</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>icpc</td>
<td>Intel C++</td>
<td>man icpc</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td><strong>Fortran Compilers</strong></td>
<td>gfortran</td>
<td>GNU Fortran95</td>
<td>man gfortran</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>ifort</td>
<td>Intel Fortran</td>
<td>man ifort</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td><strong>MPI Wrappers</strong></td>
<td>mpif77</td>
<td>MPI Fortran77</td>
<td>man mpif77</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>mpif90</td>
<td>MPI Fortran90</td>
<td>man mpif90</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>mpicc</td>
<td>MPI C</td>
<td>man mpicc</td>
<td>usq, bowery, cardiac</td>
</tr>
<tr>
<td></td>
<td>mpiCC</td>
<td>MPI C++</td>
<td>man mpiCC</td>
<td>usq, bowery, cardiac</td>
</tr>
</tbody>
</table>

Useful Information
Other compiler versions can be loaded using the `modules`

Handy Hint
Add this line before loading any modules in your batch script to declare where the "module" program is located.

For batch scripts in sh/bash

```bash
source /etc/profile.d/env-modules.sh
```

For batch scripts in csh/tcsh

```csh
source /etc/profile.d/env-modules.csh
```

Note
For online manuals and guides, please see our Finding documentation section.
Software usually needs specific settings of environmental variables. NYU HPC uses an open source software package called "Environment Modules," (or Modules for short) which allows you to add various path definitions to your shell environment. Default compilers, applications and libraries can be set by individual or combinations of Modules commands. Modules are not applications, rather they simply add the location of applications to your environment. So you can load "modules" to set up the environment which is necessary for the software.

You can list the available Modules using the command `module avail`.
<table>
<thead>
<tr>
<th>Module Path</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>/opt/modules/Modules/versions</td>
<td>3.2.6</td>
</tr>
<tr>
<td>/opt/modules/Modules/3.2.6/modulefiles</td>
<td></td>
</tr>
<tr>
<td>/opt/applications/Modules/modulefiles</td>
<td></td>
</tr>
</tbody>
</table>

```
R/5.32.00     obyss/intel/1.2.7     obyss/intel/1.3.3     obyss/intel/1.3.2
sh/pata-tga/gnu/38669     amber/wvpich/2/intel/11     amber/wvpich/2/intel/9     amber/openmpi/Intel/11
amber/18/intel-wvpich     amber/14/intel-wvpich     amber/7/intel-wvpich     ampb/gnu/1.3.8
arpack/intel/96     augustus/intel/2.5.5     automconf/2.68     automake/1.11.2
bedtools/intel/2.13.2     bedtools/intel/2.13.4     bedtools/intel/2.15.0     binutils/gnu/2.21.1a
blast/gnu/2.2.25     blast/gnu/2.2.25     blast/gnu/2.2.25     blast/intel/24
bicc/gnu/6.6.2     boost/intel/openmpi/1.44.0     boost/intel/serial/1.44.0     boost/intel/serial/1.47.0
bzip/intel/6.4.7     bowtie/Intel/0.12.7     bowtie2/intel/2.3.8-beta2     bowtie2/intel/2.3.9-beta5
breakdancer/intel/1.1-201221     bwav/intel/0.5.9     bwav/intel/0.6.1     bzip2/intel/1.0.5
bzip2/intel/1.0.5     cblas/intel/10.1.0     casava/intel/1.4.1     casava/intel/1.4.2
charm/intel/1.355/openmpi     charm/intel/1.355/serial     cmake/gnu/2.3.3     cmake/intel/4.1
comodo/4.1     comet/4.2.3     ctags/gnu/5.3     cudsock/0.4.17
ctags/gnu/5.3     cuptoolkit/gnu/4.1.17     cuflink/intel/1.1.0     cuflink/intel/1.1.0
cur/intel/7.2.23     db/intel/11.2.5.2.36     dlib/intel/2.4.1.16     dlib/gnu/1.3.9
exapro/2.2.0     exapro/intel/2.2.0     fastx/intel/2.0.1     for/gnu/2.0.0
fastq/16.0     fastq/16.0     fastx/toolkit/intel/0.0.13     [root@login-3-0 ~] #
```

The content may vary on different clusters. If you would like to run the "vmd" program, for example, you need to first find "vmd" in this list (vmd/1.8.7) and type,

```
$ module load vmd/1.8.7
```

The "module load" command loads all the configuration information for the specific software.

Then you will be able to execute "vmd" by simply typing `vmd`.

In case the module did not load or a different module was loaded by mistake, the `which` command will return an error:

```
$ which vmd
/usr/bin/which: no vmd
```

### Note
Some modules might depend on other modules.

For example, you will encounter an error message like shown below if you try to load Amber 11 directly;

```
amber11/intel-mvapich(16):ERROR:151: Module 'amber11/intel-mvapich' depends on one of the module(s) 'intel/11.1.046'
```

In this case, you have to load the intel (Intel C/C++ compiler) first.

The correct method to do so is to as follows:

```
$ module load intel/11.1.046 (as well as loading all the other prerequisite modules)
THEN
$ module load amber11/intel-mvapich
```

**Version Changes**

NYU HPC sends out notifications on a regular basis about software version upgrades. Such notifications include version number upgrade, module and path changes, as well as new features and bug fixes related to the new version.

**Requesting Additional Software**

High performance computing is a fast changing field and requires researchers to keep up with the latest developments. Experimentation is part of this process.

NYU HPC maintains a wide variety of software. Despite these efforts we cannot cover all available software. If you require additional software installations we will do our best to accommodate you. Email these requests to hpc@nyu.edu. Depending on the level of difficulty please allow for up to five working days for installation.

**Software Available at NYU**

For generic software offered by ITS, such as scp, ssh and vpn clients, please see [http://www.nyu.edu/its/software](http://www.nyu.edu/its/software)

For a complete list of all research software, including non-HPC software, please see [http://www.nyu.edu/its/research/software](http://www.nyu.edu/its/research/software)
PBS Script Generator
An interactive tool that generates PBS script based on user's input. Check [this page](#) for more details.

Front-Line HPC Consulting
HPC consultations are available once a week, Monday 1-3 PM. Appointments are required. Please make an appointment at [hpc@nyu.edu](mailto:hpc@nyu.edu).