Software Modules

The system runs RHEL 6.5. The distribution is based on gcc 4.4.7 compilers. In addition to the software packages that are part of the RHEL distribution, additional scientific software packages are available on the system and can be accessed through the 'modules' environment management tool (see http://modules.sourceforge.net/ for additional info)

The Environment Modules allow for an easy and consistent modification of the environment, needed to access specific software package avoiding conflicts and tracking correctly all dependencies.

When loading the modules for a specific package, all the environment variables gets properly modify in order to access that software. Modules can be also removed from the shell, reverting the environment to the previous state.

This helps to minimize conflicts and confusion for example when accessing different versions of the same software, or when using different software packages that in turn rely on different version of the same library.

The basic module subcommands are:

<table>
<thead>
<tr>
<th>Module subcommand</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>module available</td>
<td>Returns a list of the available software modules</td>
</tr>
<tr>
<td>module list</td>
<td>Returns a list of the modules currently loaded in the user's environment</td>
</tr>
<tr>
<td>module load &quot;some_module&quot;</td>
<td>Loads the module &quot;some_module&quot; into the current environment</td>
</tr>
<tr>
<td>module unload &lt;module name&gt;</td>
<td>Removes the module &quot;some_module&quot; from the current environment</td>
</tr>
<tr>
<td>module purge</td>
<td>Wipes the environment clean removing all the previously loaded modules</td>
</tr>
</tbody>
</table>

Example

```
[testuser@login-node-1-0 ~]$ which python
/usr/bin/python
[testuser@login-node-1-0 ~]$ module load PYTHON/ANACONDA-1.8.0
[testuser@login-node-1-0 ~]$ which python
/share/apps/scisoft/ANACONDA/1.8.0/bin/python
[testuser@login-node-1-0 ~]$ module purge
[testuser@login-node-1-0 ~]$ which python
/usr/bin/python
```