Community Software Policy

ALCF supports the deployment of community software from active projects on production systems. A project may provide and support a code on ALCF systems for the ALCF user community as described in this document. These deployments are system-specific, and their maintenance is the sole responsibility of the project deploying it. There shall be no expectation of additional support from ALCF, other than for provisioning of space and integration with the module system. Projects will be provided with an initial module file from a template, with the expectation that they will update and maintain the module, providing paths and instructions so that user communities can access the software. The user communities will get access to project-built software by way of these modules.

To support this work, ALCF Operations will create spaces in /soft/community/projects and /soft/community/modulefiles. Operations will also create a 'community' user and 'community' group to inherit ownership of projects once they reach end-of-life.

1. **Eligibility:** Existing projects with space allocated in /projects, /grand/projects, or /eagle/projects may publish software installs in /soft/community/projects and corresponding modules in /soft/community/modulefiles that are accessible through the compute resource's module system.
2. **Requesting Access:** Projects wishing to take advantage of this policy may request that a module directory be created in /soft/community/modulefiles and installation directory in /soft/community/projects by sending a request to support@alcf.anl.gov and ask to be added to the community software space. The request should include:
   - name of project (should be name of an existing project)
   - brief description
   - contact info
   - links to any additional documentation (e.g. github, bitbucket, or gitlab account, sphinx, etc.)
3. **Provisioning:** Once an access request for *project_name* is received and approved, a directory /soft/community/modulefiles/*project_name* will be created and supplied with an initial modulefile, either TCL or LUA as is appropriate for the module system. Permissions for the directory will be set rwxrwsr-x and the modulefile populated with the name, description, contact info, and links provided in the request. After the space is set up, LCF delivers a README file outlining expectation of use of the /soft/community/modulefiles/<project_name> directory, including some examples of how to point to usable builds and set appropriate permissions.
4. **Module Files:** The initially provided modulefile will be generated from a template (either TCL or LUA as is appropriate for the system) Running 'module help' for this module will return this information. As projects evolve, they are expected to update and maintain this information, which should be readily available by calling the help function for the module. Projects wishing to expose additional resources

(e.g. CI scripts, recipes, logs, examples, etc.) are encouraged to put information in the help message for their module(s).

5. **Access:** Community modulefiles will not appear in a default module path and must be loaded explicitly by first loading the 'community' module (effectively equivalent to 'module use /soft/community/modulefiles') and then loading '*project_name*'. Keeping modules out of the default path safeguards users on the system from being affected by an experimental or improperly formed community module. Loading project modules individually similarly isolates projects from disrupting use of modules provided by other projects.
6. **Project Lifetime:** When a project publishing community builds reaches the end of its lifetime, the community builds will be converted to user/group ownership as community/community, so as to give continuity to users of the software and all write permissions removed so that unmaintained installs will be read-only.