Hewlett Packard
Enterprise

HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (22.03) (S-8022)

Part Number: S-8022
Published: March 2022

HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (22.03) S-8022

Contents

1 Copyright and Version

CPE: 22.03-LocalBuild

Doc git hash: 8b0af1d94cee0d555cc95871609e2a047276c6a4

Generated: Tue Mar 01 2022

2 About the HPE CPE Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems

The *HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (S-8022)* includes procedures to install HPE Cray Programming Environment (CPE) and Parallel Application Launch Service (PALS) with HPE Performance Cluster Manager (HPCM) on HPE Cray EX and HPE Cray Supercomputer Systems.

This publication is intended for system administrators receiving their first release of this product or upgrading from a previous release. The information assumes the administrator has a good understanding of Linux system administration and HPCM.

2.1 Release Information

This publication supports installing CPE 22.03 on HPE Cray EX systems running HPCM 1.6. This release supports:

- HPE Cray EX Systems with HPCM 1.6 and SLES 15 SP2
- HPE Cray EX Systems with HPCM 1.6 and RHEL 8.4
- HPE Cray EX Systems with HPCM 1.6 and COS 2.1 or COS 2.0
- Other HPE Cray Supercomputer Systems with HPCM 1.6 and SLES 15 SP2
- Other HPE Cray Supercomputer Systems with HPCM 1.6 and RHEL 8.4
- Other HPE Cray Supercomputer Systems with HPCM 1.6 and COS 2.1 or COS 2.0
- Slurm 20.02 (or later)
- PBS Professional 2020.1.1

IMPORTANT: Use the following variable substitutions throughout the included procedures:

- <CPE_RELEASE> = `22.03`
- <CPE_VERSION> = `22.3.x`

2.2 Slurm Installation Note

The Slurm RPMs contained in the SLES HPC module do not include support for CPE; however, the Slurm RPMs provided by SchedMD include the missing components. Therefore, to use CPE with Slurm, it is necessary to download Slurm source from the SchedMD website and follow the installation instructions provided on the website.

2.3 Record of Revision

New in the CPE 22.03 publication

- Release version updates only

New in the CPE 22.02 publication

- Procedures to install independent HPE CPE RPMs were removed and will be provided when update packages are released

New in the CPE 21.12 publication

- Procedure to install the Application Task Orientation and Management (ATOM) daemon added to ***Install HPE Cray Programming Environment on SUSE Linux Enterprise Server*** and ***Install HPE Cray Programming Environment on Red Hat Enterprise Linux***
- Procedure to install the Virtual Network Identifier (VNI) daemon on a service node added to ***Install HPE Cray Programming Environment on SUSE Linux Enterprise Server*** and ***Install HPE Cray Programming Environment on Red Hat Enterprise Linux***

Publication Title	Date
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (22.03) S-8022	March 2022
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (22.02) S-8022	February 2022
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.12) S-8022	December 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.11) S-8022	November 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.10) S-8022	October 2021

Publication Title	Date
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.09) S-8022	September 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.08) S-8022	August 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.07) S-8022 Rev A	July 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.07) S-8022	July 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.06) S-8022	June 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.05) S-8022	May 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.04) S-8022	April 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.03) S-8022	March 2021
HPE Cray Programming Environment Installation Guide: HPCM on HPE Cray EX and HPE Cray Supercomputer Systems (21.02) S-8022	February 2021

3 Install HPE Cray Programming Environment on SUSE Linux Enterprise Server

PREREQUISITES

- Required and optional tar files are obtained:
  - REQUIRED: cpe-base-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
  - OPTIONAL: cpe-aocc-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
  - OPTIONAL: cpe-intel-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
  - OPTIONAL: cpe-nvidia-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
  - OPTIONAL: cpe-amd-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
- To use Lmod as the module handling system, the following must be installed in the compute image:
  - `lua53`
  - `lua53-luaposix`
  - `lua53-luafilesystem`
- To use UCX with Cray-MPICH, please note that:
  - Cray-MPICH using the UCX netmod is supported on SLES 15 SP2 systems with the HPCM installer.
  - HPE does not distribute UCX directly.
  - Mellanox provides a UCX solution as a part of their HPC-X software toolkit. This is the recommended path. Open source and Linux distro packages provide a functional, although not necessarily performant alternative.

OBJECTIVE

Install Cray Programming Environment (CPE) and optionally:

- Set Lmod as the default module handling system
- Install and configure Parallel Application Service (PALS) on systems running PBS Professional workload manager
- For systems running Slurm, see ***Slurm Installation Note***

OPTIONAL

For HPE Cray EX or HPE Cray supercomputer systems with GPU compute nodes and NOT running the Cray Operating System (COS):

- If a `rocm/x.x.x` or `cudatoolkit/x.x.x` GPU toolkit modulefile is required, refer to: https://github.com/PE-Cray/whitepapers/tree/feature/gpu/gpu-toolkit-templates, which provides environment modulefiles and pkg-config file templates.
- Systems running COS typically have GPU toolkit modulefiles pre-installed and ready for use.

IMPORTANT: Throughout this procedure, replace instances of:

- <CPE_RELEASE>
- <CPE_VERSION>

with the values specified in ***Release Information***.

PROCEDURE

1. Extract the CPE tarball.

```
admin# tar xf cpe-base-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
admin# tar xf cpe-amd-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
admin# tar xf cpe-aocc-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
admin# tar xf cpe-intel-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
admin# tar xf cpe-nvidia-<CPE_RELEASE>-sles15-sp2-hpcm-<CPE_VERSION>.tar.gz
```

2. Run the `install.sh` script.

```
admin# cpe-base-<CPE_RELEASE>-sles15-sp2/install.sh
```

Important: Repeat the previous two steps to install the optional CPE support libraries for additional compilers.

3. Enable the CPE repository.

```
admin# cm repo select cpe-<CPE_RELEASE>-sles15-sp2
```

4. Import the RPM public key.

```
admin# rpm --root /opt/clmgr/image/images/<IMAGE_NAME> \
--import cpe-base-<CPE_RELEASE>-sles15-sp2/*.asc
```

5. Install the `cm-pe-integration` and `cpe-support` RPMs, included in the CPE repository, into the image.

   ```
   admin# cm image zypper -i <IMAGE_NAME> install cm-pe-integration cpe-support
   ```

   TIP: To display available images, use the command: `cm image show`

6. (Optional) Modify `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/pe_releases` to indicate which CPE releases to install.

   The first release in the list is the default. For example, the following contents indicate that CPE 21.09 and 21.07 are to be installed with 21.09 as the default release.

   ```
   21.09
   21.07
   ```

7. (Optional) Set Lmod as the default module handling system in the image.

   TIP: An organization establishes the use of either the default CPE Module Environment system or Lmod, on a sitewide basis. The systems are mutually exclusive and cannot both run on the same system.

   a. In `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/cray-pe-configuration.csh`, change: `set module_prog = environment modules` to `set module_prog = lmod`

   b. in `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/cray-pe-configuration.sh`, change: `module_prog = environment modules` to `module_prog = lmod`

8. Install and configure the Application Task Orchestration and Management (ATOM) daemon.

   a. Install the `cray-atom` RPM into the compute node image.

   ```
   admin# cm image zypper -i <IMAGE_NAME> install cray-atom
   ```

   b. Enable `atomd` to run at startup.

   ```
   admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable atomd
   ```

9. If using the Slurm workload manager:

   a. Edit `/opt/clmgr/image/images/<IMAGE_NAME>/etc/slurm/slurm.conf` to configure Slurm to launch with CPE support by default, and to define `MpiParams`. Set:

   - `MpiDefault=cray_shasta`
   - `MpiParams=ports=20000-32767`

   ```
   admin# vi /opt/clmgr/image/images/<IMAGE_NAME>/etc/slurm/slurm.conf
   slurm.conf file generated by configurator easy.html.
   # Put this file on all nodes of your cluster.
   # See the slurm.conf man page for more information.
   #
   SlurmctldHost=slurm-host(xx.xxx.x.x)
   #
   #MailProg=/bin/mail
   MpiDefault=cray_shasta
   MpiParams=ports=20000-32767
   ...
   ```

   b. Configure Slurm to call the ATOM prolog and epilog by setting:

   - `Prolog=/opt/cray/atom/sbin/atom_slurm.py`
   - `Epilog=/opt/cray/atom/sbin/atom_slurm.py`

   or by calling:

   - `/opt/cray/atom/sbin/atom_slurm.py` from existing prolog and epilog scripts.

10. If using the PBS Professional workload manager and the system uses an HPE Slingshot NIC (Slingshot 11) network, install the Virtual Network Identifier (VNI) daemon on a service node.

    a. Install the `hpe-slingshot-vnid` RPM into the service node image:

```
admin# cm image zypper -i <SERVICE_IMAGE_NAME> install hpe-slingshot-vnid
```

b. Enable vnid to start at boot time.

```
admin# chroot /opt/clmgr/image/images/<SERVICE_IMAGE_NAME>/ systemctl enable vnid
```

c. Copy the vnid passwd file into the compute node image.

```
admin# mkdir -p /opt/clmgr/image/images/<IMAGE_NAME>/etc/vnid
admin# cp /opt/clmgr/image/images/<SERVICE_IMAGE_NAME>/etc/vnid/passwd \
/opt/clmgr/image/images/<IMAGE_NAME>/etc/vnid/passwd
```

11. If using the PBS Professional workload manager, install and configure Parallel Application Service (PALS).

a. Install the munge RPM (included in the SLES HPC module) and the cray-palsd RPM (included in the CPE repository) into the image.

```
admin# cm image zypper -i <IMAGE_NAME> install munge cray-palsd
```

b. Enable munge to run at startup.

```
admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable munge
```

c. Enable palsd to run at startup.

```
admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable palsd
```

d. Configure PALS to call ATOM. Edit /opt/clmgr/image/images/<IMAGE_NAME>/etc/sysconfig/palsd and set:

- ATOM_SOCKET=/var/run/atomd/atomd.sock

e. Enable the PBS_cray_atom hook.

```
mgmt1# qmgr -c "set pbshook PBS_cray_atom enabled=t"
```

f. If the system uses the Slingshot 11 network and vnid is installed and configured:

Edit /opt/clmgr/image/images/<IMAGE_NAME>/etc/sysconfig/palsd and set:

- PALSD_VNID_URL="https://<vnid node hostname>:5380"
- PALSD_CXI_LIB="/usr/lib64/libcxi.so.0"

12. Disable the CPE repository.

```
admin# cm repo unselect cpe-<CPE_RELEASE>-sles15-sp2
```

13. Update revision history with a comment.

```
admin# cm image revision commit -i <IMAGE_NAME> -m "Update CPE to <CPE_RELEASE>"
```

14. Activate the image for disk-less nodes only.

```
admin# cm image activate -i <IMAGE_NAME>
```

15. Verify the CPE installation.

a. Reboot one compute node.

```
admin# cm node provision -i <IMAGE_NAME> -n nid0001
```

b. Connect to the booted compute node and verify CPE modules are loaded. Note that module versions listed are examples only and may differ from those currently loaded on the system. For current CPE release product versions, see the release announcement.

```
admin# ssh nid0001
nid0001# module list
Currently Loaded Modules:
1) craype-x86-rome          5) cce/13.0.0            9) cray-libsci/21.08.1.2
2) libfabric/1.13.0.0       6) craype/2.7.13        10) PrgEnv-cray/8.3.0
3) craype-network-ofi       7) cray-dsmml/0.2.2
4) perftools-base/21.12.0   8) cray-mpich/8.1.12
```

c. For systems running PBS Professional workload manager, verify that munge and palsd services are running.

```
nid0001# systemctl status munge
   * munge.service - MUNGE authentication service
      Loaded: loaded (/usr/lib/systemd/system/munge.service; enabled; vendor prese>
      Active: active (running) since Fri 2021-01-08 20:11:37 UTC; 2 months 19 days>
        Docs: man:munged(8)
    Main PID: 2604 (munged)
     Tasks: 4
      CGroup: /system.slice/munge.service
                2604 /usr/sbin/munged
Jan 08 20:11:37 nid-1 systemd[1]: Starting MUNGE authentication service...
Jan 08 20:11:37 nid-1 systemd[1]: Started MUNGE authentication service.

nid0001# systemctl status palsd
   * palsd.service - Cray Parallel Application Launch Service Daemon
      Loaded: loaded (/usr/lib/systemd/system/palsd.service; enabled; vendor prese>
      Active: active (running) since Mon 2021-03-08 19:11:22 UTC; 3 weeks 0 days a>
    Main PID: 4109 (palsd)
       Tasks: 3
      CGroup: /system.slice/palsd.service
                4109 /usr/sbin/palsd
```

16. If the installation appears correct, proceed to reboot the system’s remaining compute nodes.

4 Install HPE Cray Programming Environment on Red Hat Enterprise Linux

PREREQUISITES

- Required tar file is obtained:
  - REQUIRED: cpe-base-<CPE_RELEASE>-rhel-8-3-hpcm-<CPE_VERSION>.tar.gz
- To use Lmod as the module handling system, the following must be installed in the compute image:
  - `lua53`
  - `lua53-luaposix`
  - `lua53-luafilesystem`
- To use UCX with Cray-MPICH, please note that:
  - Cray-MPICH using the UCX netmod is supported on RHEL 8.3 systems with the HPCM installer.
  - HPE does not distribute UCX directly.
  - Mellanox provides a UCX solution as a part of their HPC-X software toolkit. This is the recommended path. Open source and Linux distro packages provide a functional, although not necessarily performant alternative.

OBJECTIVE

Install Cray Programming Environment (CPE) and optionally:

- Set Lmod as the default module handling system
- Install and configure Parallel Application Service (PALS) on systems running PBS Professional workload manager

OPTIONAL

For HPE Cray EX or HPE Cray supercomputer systems with GPU compute nodes and NOT running the Cray Operating System (COS):

- If a `rocm/x.x.x` or `cudatoolkit/x.x.x` GPU toolkit modulefile is required, refer to: https://github.com/PE-Cray/whitepapers/tree/feature/gpu/gpu-toolkit-templates, which provides environment modulefiles and pkg-config file templates.
- Systems running COS typically have GPU toolkit modulefiles pre-installed and ready for use.

IMPORTANT: Throughout this procedure, replace instances of:

- <CPE_RELEASE>
- <CPE_VERSION>

with the values specified in ***Release Information***.

PROCEDURE

1. Extract the CPE tarball.

   ```
   admin# tar xf cpe-<CPE_RELEASE>-rhel-8-3-hpcm-<CPE_VERSION>.tar.gz
   ```

2. Run the `install.sh` script, passing the compiler environments to be installed as arguments.

   ```
   admin# cpe-<CPE_RELEASE>-rhel-8-3/install.sh <amd> <aocc> <intel> <nvidia>
   ```

3. Enable the CPE repository.

   ```
   admin# cm repo select cpe-<CPE_RELEASE>-rhel-8-3
   ```

4. Import the RPM public key.

   ```
   admin# rpm --root /opt/clmgr/image/images/<IMAGE_NAME> \
   --import cpe-<CPE_RELEASE>-rhel-8-3/*.asc
   ```

5. Install the `cm-pe-integration` and `cpe-support` RPMs, included in the CPE repository, into the image.

   ```
   admin# cm image dnf -i <IMAGE_NAME> install cm-pe-integration cpe-support
   ```

   TIP: To display available images, use the command: `cm image show`

6. (Optional) Modify `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/pe_releases` to indicate which CPE releases to install.

   The first release in the list is the default. For example, the following contents indicate that CPE 21.09 and 21.07 are to be installed with 21.09 as the default release.

```
21.09
21.07
```

7. (Optional) Set Lmod as the default module handling system in the image.

   Tip: An organization establishes the use of either the default CPE Module Environment system or Lmod, on a sitewide basis. The systems are mutually exclusive and cannot both run on the same system.

   a. In `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/cray-pe-configuration.csh`, change: `set module_prog = environment modules` to `set module_prog = lmod`

   b. in `/opt/clmgr/image/images/<IMAGE_NAME>/etc/cray-pe.d/cray-pe-configuration.sh`, change: `module_prog = environment modules` to `module_prog = lmod`

8. Install and configure the Application Task Orchestration and Management (ATOM) daemon.

   a. Install the `cray-atom` RPM into the compute node image.

   ```
   admin# cm image dnf -i <IMAGE_NAME> install cray-atom
   ```

   b. Enable `atomd` to run at startup.

   ```
   admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable atomd
   ```

9. If using the Slurm workload manager:

   a. Edit `/opt/clmgr/image/images/<IMAGE_NAME>/etc/slurm/slurm.conf` to configure Slurm to launch with CPE support by default, and to define `MpiParams`. Set:

   - `MpiDefault=cray_shasta`
   - `MpiParams=ports=20000-32767`

   ```
   admin# vi /opt/clmgr/image/images/<IMAGE_NAME>/etc/slurm/slurm.conf
   slurm.conf file generated by configurator easy.html.
   # Put this file on all nodes of your cluster.
   # See the slurm.conf man page for more information.
   #
   SlurmctldHost=slurm-host(xx.xxx.x.x)
   #
   #MailProg=/bin/mail
   MpiDefault=cray_shasta
   MpiParams=ports=20000-32767
   ...
   ```

   b. Configure Slurm to call the ATOM prolog and epilog by setting:

   - `Prolog=/opt/cray/atom/sbin/atom_slurm.py`
   - `Epilog=/opt/cray/atom/sbin/atom_slurm.py`

   or by calling:

   - `/opt/cray/atom/sbin/atom_slurm.py` from existing prolog and epilog scripts.

10. If using the PBS Professional workload manager and the system uses an HPE Slingshot NIC (Slingshot 11) network, install the Virtual Network Identifier (VNI) daemon on a service node.

    a. Install the `hpe-slingshot-vnid` RPM into the service node image:

    ```
    admin# cm image dnf -i <SERVICE_IMAGE_NAME> install hpe-slingshot-vnid
    ```

    b. Enable `vnid` to start at boot time.

    ```
    admin# chroot /opt/clmgr/image/images/<SERVICE_IMAGE_NAME>/ systemctl enable vnid
    ```

    c. Copy the vnid `passwd` file into the compute node image.

    ```
    admin# mkdir -p /opt/clmgr/image/images/<IMAGE_NAME>/etc/vnid
    admin# cp /opt/clmgr/image/images/<SERVICE_IMAGE_NAME>/etc/vnid/passwd \
    /opt/clmgr/image/images/<IMAGE_NAME>/etc/vnid/passwd
    ```

11. If using the PBS Professional workload manager, install and configure Parallel Application Service (PALS).

    a. Install the `munge` RPM (included in the SLES HPC module) and the `cray-palsd` RPM (included in the CPE repository) into the image.

    ```
    admin# cm image dnf -i <IMAGE_NAME> install munge cray-palsd
    ```

    b. Enable `munge` to run at startup.

    ```
    admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable munge
    ```

    c. Enable `palsd` to run at startup.

    ```
    admin# chroot /opt/clmgr/image/images/<IMAGE_NAME>/ systemctl enable palsd
    ```

    d. Configure PALS to call ATOM. Edit `/opt/clmgr/image/images/<IMAGE_NAME>/etc/sysconfig/palsd` and set:

    - ATOM_SOCKET=/var/run/atomd/atomd.sock

    e. Enable the `PBS_cray_atom` hook.

    ```
    mgmt1# qmgr -c "set pbshook PBS_cray_atom enabled=t"
    ```

    f. If the system uses the Slingshot 11 network and `vnid` is installed and configured:

    Edit `/opt/clmgr/image/images/<IMAGE_NAME>/etc/sysconfig/palsd` and set:

    - `PALSD_VNID_URL="https://<vnid node hostname>:5380"`
    - `PALSD_CXI_LIB="/usr/lib64/libcxi.so.0"`

12. Disable the CPE repository.

    ```
    admin# cm repo unselect cpe-<CPE_RELEASE>-rhel-8-3
    ```

13. Update revision history with a comment.

    ```
    admin# cm image revision commit -i <IMAGE_NAME> -m "Update CPE to <CPE_RELEASE>"
    ```

14. Activate the image for disk-less nodes only.

    ```
    admin# cm image activate -i <IMAGE_NAME>
    ```

15. Verify the CPE installation.

    a. Reboot one compute node.

    ```
    admin# cm node provision -i <IMAGE_NAME> -n nid0001
    ```

    b. Connect to the booted compute node and verify CPE modules are loaded. Note that module versions listed are examples only and may differ from those currently loaded on the system. For current CPE release product versions, see the release announcement.

    ```
    admin# ssh nid0001
    nid0001# module list
    Currently Loaded Modules:
    1) craype-x86-rome          5) cce/13.0.0            9) cray-libsci/21.08.1.2
    2) libfabric/1.13.0.0       6) craype/2.7.12        10) PrgEnv-cray/8.1.0
    3) craype-network-ofi       7) cray-dsmml/0.2.2
    4) perftools-base/21.12.0   8) cray-mpich/8.1.12
    ```

    c. For systems running PBS Professional workload manager, verify that `munge` and `palsd` services are running.

    ```
    nid0001# systemctl status munge
       * munge.service - MUNGE authentication service
          Loaded: loaded (/usr/lib/systemd/system/munge.service; enabled; vendor prese>
          Active: active (running) since Fri 2021-01-08 20:11:37 UTC; 2 months 19 days>
            Docs: man:munged(8)
        Main PID: 2604 (munged)
         Tasks: 4
          CGroup: /system.slice/munge.service
                   2604 /usr/sbin/munged
    Jan 08 20:11:37 nid-1 systemd[1]: Starting MUNGE authentication service...
    ```

```
Jan 08 20:11:37 nid-1 systemd[1]: Started MUNGE authentication service.

nid0001# systemctl status palsd
   * palsd.service - Cray Parallel Application Launch Service Daemon
      Loaded: loaded (/usr/lib/systemd/system/palsd.service; enabled; vendor prese>
      Active: active (running) since Mon 2021-03-08 19:11:22 UTC; 3 weeks 0 days a>
    Main PID: 4109 (palsd)
       Tasks: 3
      CGroup: /system.slice/palsd.service
                └─4109 /usr/sbin/palsd
```

16. If the installation appears correct, proceed to reboot the system’s remaining compute nodes.

5 Create Modulefiles for Third Party Products

PREREQUISITES

- Third-party packages are downloaded and installed

OBJECTIVE

These instructions use `crypkg-gen` to create a modulefile for a specific version of a supported third-party product. This allows a site to set a specific version as default.

The following tasks are necessary and can be embedded in a script where a third-party product is being installed.

PROCEDURE

1. Load `craypkg-gen` module.

   ```
   admin# source /opt/cray/pe/modules/admin# default/init/bash
   admin# module use /opt/cray/pe/modulefiles
   admin# module load craypkg-gen
   ```

2. Generate module and set default scripts for products. Where:

   AMD Optimizing C/C++ Compiler: (requires `craypkg-gen` >= 1.3.16)

   ```
   admin# craypkg-gen -m /opt/AMD/aocc-compiler-<MODULE_VERSION>/
   ```

   Nvidia HPC SDK (requires `craypkg-gen` >= 1.3.16)

   ```
   admin# craypkg-gen -m /opt/nvidia/hpc_sdk/Linux_x86_64/<MODULE_VERSION>/
   ```

   Intel oneAPI

   ```
   admin# craypkg-gen -m /opt/intel/oneapi/compilers/<MODULE_VERSION>/
   ```

   AMD ROCm

   ```
   admin# craypkg-gen -m /opt/rocm-<MODULE_VERSION>
   ```

3. Run a `set default` script.

   ```
   admin# /opt/admin-pe/set_default_craypkg/set_default_<MODULE_NAME>_<MODULE_VERSION>
   ```

5.1 Create a Custom Dynamic Hierarchy

PREREQUISITES

- Lmod is set as the default module handling system

OBJECTIVE

For the CPE Custom Dynamic Hierarchy to detect the desired Lmod module path, one or more custom dynamic environment variables must be created according to the requirements defined within this procedure.

PROCEDURE

To create a custom dynamic environment variable:

1. The environment variable name begins with `LMOD_CUSTOM_`.
2. Append the descriptor of the module type that the environment variable will represent. The module types and descriptors are:

Module Type	Descriptor
Compiler	`COMPILER_`
Network	`NETWORK_`
CPU	`CPU_`
MPI	`MPI_`
Compiler/Network	`COMNET_`
Compiler/CPU	`COMCPU_`
Compiler/Network/CPU/MPI	`CNCM_`

   Example: The custom dynamic environment variable for the combined compiler and CPU module begins with `LMOD_CUSTOM_COMCPU_`.

3. Following the descriptor, append all prerequisite module aliases along with their respective compatible versions. See *Module Path Aliases and Current Compatibility Versions*. The format of the module path alias/compatible version string for each module type is shown below. Note that due to publishing issues, long module alias/compatible version strings are split across two lines as indicated below.

   Module Type: Module Path Alias/Compatible Version String

   Compiler: <compiler_name>/<compatible_version>

   Network: <network_name>/<compatible_version>

   CPU: <cpu_name>/<compatible_version>

   MPI: String definition is split across two lines

   <compiler_name>/<compatible_version>/<network_name>/<compatible_version>/

   <mpi_name>/<compatible_version>

   Compiler/Network: <compiler_name>/<compatible_version/<network_name>/<compatible_version>

   Compiler/CPU: <compiler_name>/<compatible_version>/<cpu_name>/<compatible_version>

   Compiler/Network/CPU/MPI: String definition is split across two lines

   <compiler_name>/<compatible_version>/<network_name>/<compatible_version>/

   <cpu_name>/<compatible_version>/<mpi_name>/<compatible_version>

   TIP: To create an acceptably formatted environment variable name, all slashes and dots in the module alias/compatible version string must be replaced with underscores and all letters must be uppercase.

   Example Module Path Alias/Compatible Version Strings

   - **Compiler** = `cce`

     The path alias/compatible version string (values found in *Module Path Aliases and Current Compatibility Versions*) is `crayclang/10.0`; therefore, the text added to the environment variable name is `CRAYCLANG_10_0`.

- **Network** = `craype-network-ofi`

  The path alias/compatible version string is `ofi/1.0`; therefore, the environment variable text is `OFI_1_0`.

- **CPU** = `craype-x86-rome`

  The path alias/compatible version string is `x86-rome/1.0`; therefore, the environment variable text is `X86_ROME_1_0`.

- **MPI** = `cray-mpich`

  `cray-mpich` has two prerequisite module types (compiler and network). Therefore, the environment variable must include the alias/compatible version for the desired compiler, network, and MPI. For a `cray-mpich` module dependent on `cce` and `craype-network-ofi` the path alias/compatible version string is `crayclang/10.0/ofi/1.0/cray_mpich/8.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0_CRAY_MPICH_8_0`.

- **Compiler/Network** = `cce` with `craype-network-ofi`

  The path alias/compatible version string is `crayclang/10.0/ofi/1.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0`.

- **Compiler/CPU** = `cce` with `craype-x86-rome`

  The path alias/compatible version string is `crayclang/10.0/x86-rome/1.0`; therefore, the environment variable text is `CRAYCLANG_10_0_X86_ROME_1_0`.

- **Compiler/Network/CPU/MPI** = `cce`, `craype-network-ofi`, `craype-x86-rome`, and `cray-mpich`

  The path alias/compatible version string is `crayclang/10.0/ofi/1.0/x86-rome/1.0/cray-mpich/8.0`; therefore, the environment variable text is `CRAYCLANG_10_0_OFI_1_0_X86_ROME_1_0_CRAY_MPICH_8_0`.

4. Append the text: `_PREFIX` following the final module/compatibility text instance. Creation of the custom dynamic environment variable is now complete.

   Example: Network = `craype-network-ofi`

   The custom dynamic environment variable is `LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX`.

NEXT: Add the custom dynamic environment variable to the user environment by exporting it with its value set to the Lmod module path.

Example: Network = `craype-network-ofi`

After executing the command below, all modulefiles in `<lmod_module_path>` are shown to users whenever `craype-network-ofi` is loaded

```
# export LMOD_CUSTOM_NETWORK_OFI_1_0_PREFIX=<lmod_module_path>
```