Apex Cluster Manual

v2.2

Table of Contents

Introduction

Welcome

Thank you for your purchase of an Advanced Clustering Technologies' Apex Cluster!

This manual will provide administrators and users with information required for the management and operation of a computing cluster. Every attempt has been made to provide accurate information and instructions. As each Apex Cluster is custom built and configured the instructions contained within may not be completely accurate. In most instances, deviations from this manual will be noted on the accompanying documentation.

Intended Audience

This manual provides information that will be applicable to system administrator, support technicians and end users, essentially anyone that will access the Apex Cluster.

Quick Reference

Contacting Advanced Clustering

Every attempt has been made to provide thorough information in this manual. We realize that not all questions can be anticipated. Fortunately, every Apex Cluster comes with lifetime software support and at least a one year hardware warranty. Longer warranty options are available through your sales representative.

When contacting Advanced Clustering for technical support reasons please have the serial number of the computer or device in question.

We provide many methods for you to contact our technical support specialists.

Email	support@advancedclustering.com
Web	http://www.advancedclustering.com/support/
Phone	1 (866) 802-8222 (toll free)
Phone	1 (913) 643-0300 Option 2 (outside of North America)
Fax	1 (913) 643-0299

Default Passwords

This list includes the default passwords for the item commonly contained within an Apex cluster. For security purposes, these passwords should be changed by the administrator of the Apex Cluster.

Device	Username	Password	Notes
All nodes	root	cluster	
All nodes	act	cluster	Used for testing while being configured

Cluster Overview

Any time two or more computers are used on one problem it can be described as a cluster. Apex Clusters are High Performance Computing (HPC) clusters. This type of clustering uses multiple computers to work on one problem. When an HPC cluster is created using "off the shelf" hardware and open source software it is sometimes referred to as a Beowulf cluster.

Another common type of clustering is called High Availability (HA). These clusters provide resiliency for applications that need enhanced uptime. These clusters are not addressed in this manual.

Node Roles

The computers that make up a cluster are called nodes. Due to the open nature of HPC clusters there may be many different node types in a single cluster. Most Apex Clusters are comprised of two types of nodes: head nodes and compute nodes. Sometimes head nodes may be referred to as front-end or interactive nodes. These are functionally equivalent to a head node.

Head Node

Most clusters have one head node. This node is used for all user interaction. Software services required for cluster operation originate from the head node but the compute jobs themselves do not typically run on the head node. The common tasks assigned to a head node are:

- User log ins
- Router and firewall to an external network
- Compiling and debugging software
- File serving
- *Queuing* or job scheduler
- Web server for management tools
- InfiniBand Subnet Manager (If the head node has IB)

Compute Nodes

Compute nodes are used for the computational work. Logging in to compute nodes is discouraged as that can interfere with other users' jobs.

Once booted, compute nodes will mount any filesystems shared from the head node and start the execution piece of the queuing system. At this point the compute node is ready to accept and run jobs from the queuing system.

Network Layout

The cluster communicates on a private network such as 192.168.1.0/24. The head node will provide and additional link to an external network and act as a router, or gateway, to the external network. The use and configuration of the external network interface is site specific. You will need to contact your institution's network administrators for the information required.

Hostname resolution within in the cluster is handled with a distributed *hosts* file. This file is named */etc/hosts* and resides on every node in the cluster. To redistribute the */etc/hosts* file after a change please see the section titled “Using Act_utils” on page 26.

Shared Filesystems

The head node of the cluster will have at least two filesystems shared with the compute nodes in the cluster. The path to the files contained in these directories is consistent on every node of the cluster.

/home will contain each user's home directory. This directory provides a place for each user to store their data and software.

/act contains software, documentation and example files installed by Advanced Clustering Technologies. Do not delete this directory as it contains the files needed to recover your cluster in the event of a failure.

Node Communication

Communication between the nodes in the cluster are handled by **`ssh`** or **`rsh`**. Sometimes both are enabled. All nodes allow logins from all other nodes without the need for a password.

Cluster Hardware

Unpacking Your Cluster

Weight Warning

WARNING: The components of an Apex Cluster can be very heavy. Some pre-assembled clusters are over 2000 lbs. Great care must be used when unpacking and/or moving a cluster or any of its components.

Check For Shipping Damage

Despite all attempts at safe and secure packaging, occasionally some packages get damaged during shipping.

Any suspected shipping damage MUST be reported WITHIN 48 HOURS. After 48 HOURS we will be unable to file any claims with the carrier or the insurer.

Tools Required

Installing an Apex Cluster will require the following common tools.

- Box Cutter
- Phillips head screwdriver
- Slot head screwdriver
- Diagonal cutters (also called wire cutters)
- 1/2" wrench or socket
- (Optional) drill or electric screwdriver

Unpacking A Rack

Unpacking a rack will require a minimum of three people. Do not attempt to unpack a rack alone.

The following tools will be used:

- Box cutter
- 1/2" wrench or socket
- Diagonal cutters

If you have a drill or electric screw driver with a socket attachment, it can be used to remove the bolts securing the rack to the pallet.

Do not discard the pallet or the packaging for the rack. In the event that your Apex cluster may need to

be moved these packing materials are required for the safe transit of the rack.

- Use the box cutter to cut the plastic stretch wrap. Take care to not cut through to the thicker plastic covering placed directly over the rack. Remove the stretch wrap from the rack.
- Remove the cardboard corner protectors.
- Remove the documentation packet from the plastic covering.
- Slide the plastic covering up and off of the rack.
- Open both sets of doors on the rack and cut the zip tie that attaches the key to the inside of the door. Keep these keys in a safe place. The keys are required for the doors and side panels.

Removing And Attaching Doors And Side Panels

There are some situations that may require the rack doors and/or side panels of the rack to be removed like installing new equipment or reducing weight when moving a rack.

Removing A Door

- Open door as far as it will open.
- Disconnect the grounding wire on the inside hinge side of the door. There is a spade connector in the middle that should pull apart easily.
- Gently lift up on the door

Attaching A Door

- Lift the door and place the hinges in matching grooves.
- Lower the door into the holes making sure that all hinges are firmly in position.
- Reconnect the grounding wire by pushing the male end of the spade connector into the female end.

Removing A Side Panel

- Use the rack keys to unlock the side panel.
- Pull the siding latch directly below the key hole.
- Tilt the side panel away from the rack and lift up to remove.

Attaching A Side Panel

- Slide the bottom portion of the side panel into the ridge on the rack.
- Tilt the top of the side panel up until it latches securely into the rack.
- Lock the side panel using the key.

Shock Pallet

Fully assembled racks can weigh up to 2000 pounds! Use extreme caution unpacking and moving.

Fully assembled Apex Clusters are shipped on shock pallets. These pallets have a layer of shock

absorbing foam built in to protect the rack, and its contents, from shock and vibration. Also included are ramps to remove the rack from the pallet. If the ramps are missing do not attempt to remove the rack from the pallet by other means; Immediately contact Advanced Clustering (see page 7 for information) to obtain replacement ramps.

- Using the box cutter, carefully cut the plastic stretch wrap and remove it from the rack.
- The ramps are covered in plain cardboard. Remove them from the side of the rack and remove the cardboard.
- Locate the holes on the top surface of the pallet and place the metal tabs of the ramps in to holes. The ramps must be fully inserting and align flush with the top of the pallet.
- There are brackets located along the bottom edge of the rack on the front and back sides. Use the 1/2" wrench or socket to remove the bolts and brackets. Retain these bolts and brackets.
- Do not push the rack from the top, it can tip over! Slowly roll the rack toward the ramps taking care to line the casters up in the middle of the ramp.
- Carefully roll the rack down the ramps.

Standard Pallet

A rack with rails and cables can be up to 500 pounds! You will need at least three people to remove a rack from a standard pallet.

Apex Clusters that are to be assembled on site are shipped on standard packing pallet. The rack will have the sliding rails, cables and other accessories installed.

- To reduce weight remove the doors and side panels. Instructions are located on page 11.
- There are brackets located along the bottom edge of the rack on the front and back sides. Use the 1/2" wrench or socket to remove the bolts and brackets. Retain these bolts and brackets.
- Carefully lift the rack then slide the pallet from underneath.
- Gently place the rack down on the floor.

Releasing Cables Tied For Shipment

To prevent damage during shipping, some of the pre-installed cables will be bundled up and secured to the rack. These will be attached with black, white, or green zip ties that have not been trimmed. There will be a long "tail" hanging from the zip tie.

To release the cables from the zip tie, use the diagonal cutters to cut the zip tie. Be careful to avoid cutting the cables and cut the zip tie between two cables.

Unpacking Nodes

Unpacking Apex Cluster nodes will require at least two people. A box cutter will also be required.

Please retain at least one box for each type of node. In the event that warranty service is needed, the original box will be required to protect the node from damage.

- If the nodes are on a pallet they will be covered in plastic stretch wrap. There may also be nylon straps holding the boxes to the pallet. Using a box cutter, carefully cut any straps and remove

the stretch wrap.

- Working with one node at a time, remove a box from the stack on place it on the ground. Cut the tape sealing the box with a box cutter. Take care to not cut too deep.
- Open the inner box.
- Lift the cardboard panel out of the box.
- With one person per side lift the node up and out of the inner box.

1U Blade Assembly

NOTE: Installation of the 1U Blade chassis is best performed with two people.

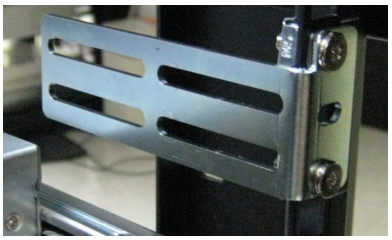
Installing Blade Housing

- Open the hardware packet. It will contain 10 cage nuts, 8 long screws, and 4 short screws. The 2 rear ears are also shown in this picture.
- Install 2 cage nuts in the rear of the rack. Use the top and bottom holes in the rack space

- Install 2 cage nuts in the front of the rack

- Attach the rear ear to the cage nuts using the long screws. Please note the orientation of the ear in the photo. The protruding mounting hole should be placed toward the top.

- This picture provides another view of the rear ear.



- Remove both blades from the outer housing.
- Slide the outer housing into the rack and align the rear ears with the channels in the housing. Continue to support the front of the housing.
- Secure the front of the outer housing using 2 screws per side.

Inserting a Blade

- Pay careful attention to the fans while inserting the blade. The vibration reducing mounting of the fans can cause then to catch on the housing.
- Line up the blade with the opening in the outer housing.
- Gently slide the blade fully into the housing.

Removing a Blade

- Remove all cables connected to the back of the blade. Power, Ethernet, and InfinBand cables could be connected and could be damaged.
- Grab both front latches and squeeze them toward the middle of the blade.
- While still squeezing the latches, gently pull the blade out of the housing.

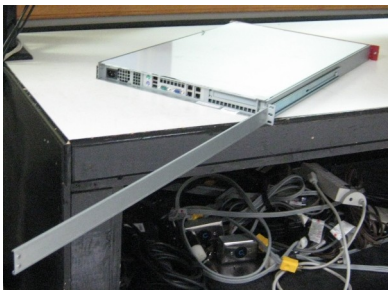
1U Rail Assembly

NOTE: These instructions are for the rails supplied with 1U chassis beginning in March 2008. These rails are identified by a purple latch on the piece attached to the computer.

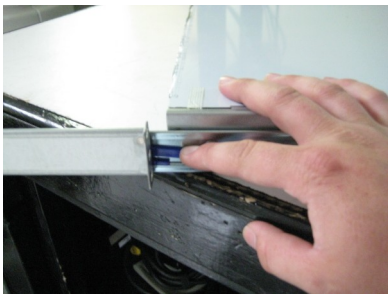
- Open the hardware packet. It will contain 10 cage nuts, 8 long screws, and 4 short screws. The 2 rear ears are also shown in this picture.

- The front piece of the rail is on the computer.

- Fully extend the front rail piece on the computer.



- Press the purple release tab down to release the front rail piece.



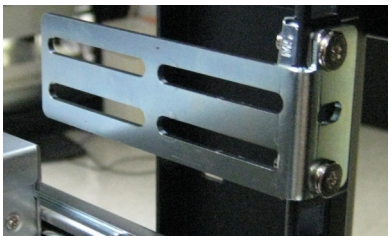
- Remove the front rail piece from the computer.

- Install 2 cage nuts in the rear of the rack. Use the top and bottom holes in the rack space.

- Install 3 cage nuts in the front of the rack.

- Attach the rear ear to the cage nuts using the long screws. Please note the orientation of the ear in the photo. The protruding mounting hole should be placed toward the top.

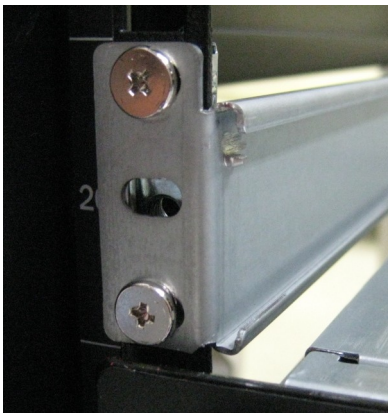
- This picture provides another view of the rear ear.



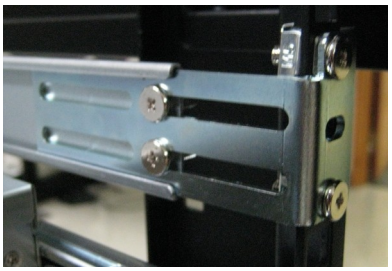
- Slide the front rail onto the rear ear.

- This picture shows the rear ear completely inserted.

- Attach the front of the rail with 2 long screws. Use the top and bottom holes.



- Use 2 short screw to secure the front rail piece to the rear ears.



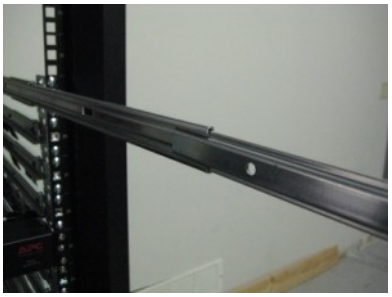
Inserting 1U Rails

Inserting a 1U sized computer into a rack will require two people: one person per side. Do not let go of the server until it is firmly locked in to the rail.

- Fully extend the portion of the rail installed in the rack.

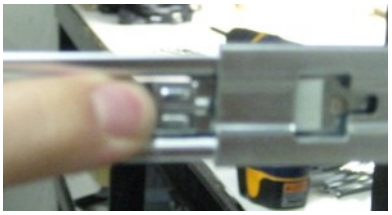
- Line up the rail piece attached the the computer with the piece attached to the rack.

- Insert the computer piece of the rail. Take care to keep both sides even.

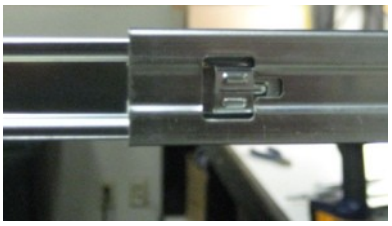


- The rail lock may catch on the outer casing of the rail.

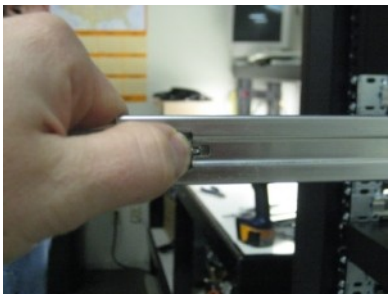
- Carefully depress the lock and slowly slide the computer the rest of the way into the rail.



- The rail will lock in.



- To fully insert the computer the rail lock must be depressed.



Inserting 2U and Larger Rails

Computers that are larger than 1U in size use a different rail. These computers are also typically heavier and may require three people to insert into a rack.

The steps for inserting 2U and larger computers is similar to Inserting 1U Rails with the following exceptions.

- When extending the rail make sure that the gray piece with ball bearings is slid to the very front of the rail. The computer side of the rails fits in the channels created by the ball bearings.

- The locking mechanism is a small black or silver lever.

- The lever on the left side will be pushed down; the lever on the right side will be pushed up.

Inserting A Disk Array

The rack mounting rails used for disk arrays will vary by model. Some work like the 2U+ rails while others are simply L shaped brackets that the disk array rests upon. The instructions for disk arrays are very general guidelines. A populated disk array can be 150 lbs. and should not be lifted by one person.

To reduce weight, the drives can be removed from a disk array when it is powered off. The position of the drives must be noted to avoid corrupting the array. Once the drives are removed the array can be placed into the rack and the drives re-inserted.

Inserting A Switch

Most Ethernet, serial console, and KVM switches mount directly to the frame of the rack using standard screws. The screws are installed in the rack, in the appropriate positions and will need to be

removed before mounting the switch. Two people are required for mounting switches. One person will hold the switch in place while the other installs the screws.

Some large switches, such as InfiniBand and high port count gigabit Ethernet, will have a shelf or L-shaped brackets. These switches are heavy and/or awkward to handle.

Connecting Cables

The cables installed in an Apex Cluster are precisely fit to reach to appropriate connector. Excess cable is neatly bundled to the sides allowing maximum airflow through the rack and a neat appearance that facilitates easy access in the event of trouble.

Do not make sharp bends in cables or pull hard on the cables. The cables will easily reach their destination. InfiniBand cables are especially sensitive to bending and are relatively expensive to replace.

Ethernet

Ethernet, serial, and KVM cables may all use CAT5 or CAT6 cable (Ethernet cable). By default the following color scheme is used to denote for which function a cable is used.

Color	Function
Red	Serial console or IPMI
Black	Ethernet for computers
Yellow	Ethernet for power control devices
Blue	KVM

InfiniBand Cables

InfiniBand cables are easily damaged by excessive bending!

InfiniBand cables have a latch mechanism on them to prevent the accidental removal of a cable. When connecting a cable make sure that this latching mechanism fully engages to ensure a firm connection.

The latching mechanism can make removing InfiniBand cables difficult or tedious.

- The latch release will be one of the following:
    - A blue tab that can be pulled
    - A gray tab that can be pulled
    - A white loop that can be pulled
- With one hand push the connector firmly into the port to which it is connected
- With the other hand pull the release tab or loop
- While holding the release tab, use the first hand to grab the body of the connector and pull straight out
- Any difficulty in removing the cable indicates that the latch has not been released. Start at the beginning of these instructions.

Startup and Shutdown

Powering On A Cluster

The devices in an Apex Cluster should be powered on in the order listed below. Make sure that the devices in each step are fully booted before moving to the next step.

- Power distribution devices including uninterruptible power supply and network controlled power switches
- Ethernet switches, InfiniBand switches, serial consoles and KVM switches
- External disk arrays and tape drives
- Storage sub-clusters (Luster, PVFS2, Panasas) if included
- The cluster head node
- Any miscellaneous management nodes
- Compute nodes

If a device does not come on when plugged in look for "1/0" printed on a switch or next to a button. Switches should be pushed to the "1" side and buttons should be depressed fully. If a device fails to power on contact Advanced Clustering (pg. 7) immediately.

Powering Off A Cluster

Like powering on a cluster (pg. 23), powering off a cluster should be performed in a specific order. Having a working Act_utils installation will aid in the process of powering off the cluster. Please see the section titled “Using Act_utils” (pg. 26) for more information.

Make sure the nodes or devices at each step are fully powered off before moving to the next step.

- Shut down the compute nodes. If Act_utils is configured with a group named "nodes" for the compute nodes the following command can be used:
  `act_exec -g nodes poweroff`
- Turn off any miscellaneous management nodes such as log in nodes. If Act_utils is configured with a group for these management nodes, substitute that group name :
  `act_exec -g [`*`group name]`* **`poweroff`
- Shut down the head node:
  `/sbin/shutdown -h now`
- Shut down the storage sub cluster, if included.
- Shut down any external disk arrays or tape drives.
- Unplug any Ethernet switches, InfiniBand switches, serial consoles or KVM switches.
- Unplug power distribution devices including uninterruptible power supply and network controlled power switches.

User Management

Adding and deleting users is one of the most common administrative tasks performed on clusters. All of the commands given should be run on the head node by the *root* user.

/etc/skel

The directory, */etc/skel*, is a "skeleton" home directory. Any files or directories in */etc/skel* get copied to a new user's home directory when that user is created. Most of the files contained in */etc/skel* are called “dotfiles” as they are prefaced with dot, or period (.). These files do not show up when listing the directory contents with the *ls* command unless the **`-a`** option is given. Example: **`ls -a /etc/skel`**

act.sh and .actrun

/etc/skel contains a file name *.actrun* . When this file is present in a user's home directory */etc/profile.d/act.sh* will prompt the user to create some files in their home directory that are used when running parallel programs. If the user denies the creation of any of the files a message will be inserted into the system log. Please note that the *.actrun* file will be deleted once the configuration is complete.

Adding A User

Users are added with the **`useradd`** command. Replace **`<username>`** with the desired username.

`useradd <username>`
This will create the user, their group and copy the contents of */etc/skel* to */home/<username>* .

Next, a password will be assigned to the user using the **`passwd`** command. You will be prompted to enter the new user's password twice.

```
[root@head root]# passwd [username]
Changing password for user [username].
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Deleting A User

Users are deleted with the **`userdel`** command. By default, **`userdel`** leaves the user's home directory in tact. Adding the options **`-r`** and **`-f`** will force the removal of all files in the user's home directory.

`userdel <username>`

Synchronizing Users

Proper operation of a cluster requires that all users be present on all nodes. Apex clusters support two methods of synchronization: **act_authsync** and NIS. **act_authsync** is the preferred method.

Synchronizing With act_authsync

act_authsync, when run on the head node, will copy the head node's authentication files to the nodes. Example: **act_authsync -a**

For more information on **act_authsync** and the other Act_utils software please see page 26.

Updating NIS Maps

If your cluster is using NIS the NIS *maps* will need to be updated after adding or deleting a user.

```
[root@head root]# cd /var/yp
[root@head root]# make
```

Using Act_utils

Act_utils is a collection of programs that aid in the administration of a cluster. *Act_utils* is only available with an Apex Cluster from Advanced Clustering Technologies.

Act_utils replaces *Beo_utils*.

Act_utils Configuration

All configuration for *Act_utils* is contained in the */act/etc* directory. This section provides an overview of the function of the various configuration files. Each file is commented as to its specific format and possible options.

act_util.secret

act_util.secret is only readable by the root user. This file is used to store passwords for some devices that will be managed via *Act_utils*. Commonly, IPMI passwords and SNMP write communities are stored in *act_util.secret*.

This file uses a sectioned INI format with a section name in square brackets, ***[section name]***, and data stored in key/value pairs in the format of ***key=value***

They values in this file are referenced in *act_util.conf* via the following syntax:

And example section that defines username and password information for an IPMI interface:

```
[ipmi]
username=super
password=cluster
```

To reference these values in act_utils.conf:

```
ipmi_username=<ipmi:username>
ipmi_password=<ipmi:password>
```

act_util.conf

act_utils.conf defines all of the devices in the cluster: nodes, switches, IPMI interfaces, power control devices, etc. This file uses a sectioned INI format. Sections are denoted with square brackets, ***[section name]***. Each section contains key/value pairs in the format ***key=value***. The default file shipped with *Act_utils* is heavily commented and shows examples of each type of section.

Section names must be unique, alphanumeric, and can not contain spaces or special characters. Other than the few restrictions, sections can be named anything that makes sense to the configuration of the cluster.

There are seven types of sections: **config**, **host**, **range**, **group**, **pdu**, **console**, and **device**.

type=config

The **config** section, of which there must be one and only one, is the global configuration for all of the tools in *Act_utils*.

- **exec_cmd** - The command that is used to execute commands on the remote machines. The default is **ssh**. **rsh** is also supported.
- **cp_cmd** (required) - The command that is used to copy files to the remote machines. The default is **scp**. **rcp** is also supported.
- **domainname** (optional) - The domain name of the machines in the cluster.
- **netmask** - The netmask of the cluster's private network.
- **gateway** - The gateway, or route, for the cluster's private network. This is usually the private address of the head node.
- **modelfile_path** (required) - The directory that contains configuration data for Advanced Clustering products.
- **secretfile** (optional) - The path to the *act_util.secret* (pg. 26), a file that is only accessible by root that stores passwords to be used by the *Act_utils* commands.
- **nodefile** - The path to *act_nodes.conf* (pg. 30) while contains details about nodes such as serial number, mac addresses, console ports, and power control ports.

Network Configuration

As of version 2 of *Act_utils*, multiple network interfaces can be specified in *act_utils.conf*.

All options for networking start with **dev[*device_name*]_** . ***device_name*** can be any valid network device available on the system; for example: **eth0**, **ib0**, or **bond0**. The only reserved device name is **ipmi**.

- **bootproto** - The boot protocol for the network interface. Can be **none** or **dhcp**.
- **onboot** - yes or no. Specifies if the interface should be started at boot.
- Address Specification:
  - **ipaddress** (for single host defintion) - The IP address for the interface.
  - **ipstart** / **ipend** (for host range definition) - The starting and ending IP address for a range of nodes.
- **gateway** - Gateway address.
- **netmask** - Netmask of the network.
- Hostname specification:
  - **hostname** (for single host definition) - The hostname associated to the address of this interface. Must be unique for each interface.

- **hostpad** (for host range definition) - Padding for the numbers used in hostname. A value of 2 will create ***02***, ***22***, ***122***; A value of three will create ***002***, ***022***, ***122***.
- **hostprefix** / **hostsuffix** (for host range defintion) - Used to assemble the hostname for range. The padded node number is placed between the hostprefix and hostsuffix. hostsuffix is not required. Must be unique for each interface.

IPMI Interface

ipaddress, **ipstart**, **ipend**, **gateway**, **netmask**, **hostname**, **hostpad**, **hostprefix**, and **hostsuffix** operate the same as a standard network interface. The *ipmi* device adds the following fields:

- **username** - The user name used to connect to the IPMI device.
- **password** - The password used to connect to the IPMI device.

For security reasons, it is advised to not store the username and password values in *act_util.conf*. These should be stored in *act_util.secret (see pg. 26).*

Bonding

ipaddress, **ipstart**, **ipend**, **gateway**, **netmask**, **hostname**, **hostpad**, **hostprefix**, and **hostsuffix** operate the same as a standard network interface. The option **bondslaves** is a comma separated list of network devices to be included in the bond.

The bonds slave devices include **onboot**, **bootproto**, and **bondmaster**, which refers to the band master device.

type=host

The *host* type defines a single host in the cluster. This is most often used for head nodes and storage nodes. Hosts that are named in a sequential fashion, such as compute nodes, are best configured in a range. The host type uses the network options specified on pg. 27. The following options are specifc to the host type.

- **modelfile** (optional) - The Advanced Clustering product definition file to be used for this host.
- **cloner_image** - Name of the cloner image for this host.
- **default_dev** - Default network device.
- **gateway_dev** - Device to use for the default gateway if the **default_dev** and desired gateway are on different network devices.

type=range

Range defines a series of sequentially named nodes; this is typically used for compute nodes. The range type uses the network options specified on pg. 27.

- start - The starting number for the sequential hosts in the range.

- end - The ending number for the sequential hosts in the range.
- **`modelfile`** (optional) - The Advanced Clustering product definition file to be used for this host.
- **`default_dev`** - Default network device.
- **`gateway_dev`** - Device to use for the default gateway if the **`default_dev`** and desired gateway are on different network devices.

type=pdu

PDU, or power distribution units, allow network control of their electrical outlets. These are used to turn devices on or off.

- **`module`** - The module defines what type of PDU this device is. Currently supported modules are **`apc_rPDU`** for current APC PDUs, and **`apc_masterswitch`** for the older APC Masterswitch devices.
- **`community`** - This specifies the SNMP write community that is used to control the outlets on the PDU. For security, this value should reference the *act_util.secret* file (pg . 26).
- **`ipaddress`** - The IP address of the PDU.

type=console

This type defines a serial console device.

- **`module`** - The module defines the method used to connect to the console. Currently, only **`ssh_port`** is supported.
- **`username`** - The user name to be used to connect to the console device. This should reference the *act_util.secret* file (pg . 26).
- **`password`** - The password to be used to connect to the console device. This should reference the *act_util.secret* file (pg . 26).
- **`ssh_start_port`** - The starting address of the console ports as accessible via SSH. For OpenGear consoles, this is 3001.
- **`ipaddress`** - The IP address of the console device.

type=device

A generic network device.

- **`ipaddress`** - The IP address of the device.

type=group

A group can include any hosts or ranges that are defined in the file.

- **include -** A comma separated list of hosts or ranges to be included in the group. Spaces are not allowed.

act_nodes.conf

act_nodes.conf stores specific data about nodes: hostname, Advanced Clustering serial number, Ethernet MAC addresses, power control ports, and console server ports. Apex clusters are shipped with a complete *act_nodes.conf*.

Each line contains the following items separated by white space:

- Host name
- Advanced Clustering serial number
- Ethernet device names and MAC addresses. Format: ***device1=macaddr,device2=macaddr*** No spaces are allowed in the string. There is no limit to the number of network devices.
- PDU device and port. Format: ***device1:port,device2:port*** The device is a PDU as defined in *act_util.conf* (pg. 26). The port is electrical outlet port number to which the node is connected. Multiple devices and ports can be used for nodes with redundant power supplies. No spaces are allowed in the string. There is no limit the the number PDUs and ports.
- Console server and port: Format **device:port** The device is a console as defined in *act_util.conf (pg. 26)*. The port is the console port to which the device is connected.

An example showing a head node with multiple PDU ports and a console port, node01 with a console port, and node02 with a PDU port.

```
head   325128 eth0=00:15:17:26:87:de,eth1=00:15:17:26:87:dc pdu1:1,pdu2:1 console1:48
node01 399999 eth0=00:1f:c6:0c:d4:0a,eth1=00:1f:c6:0c:d4:0b none          console1:1
node02 324757 eth0=00:23:54:19:a1:43,eth1=00:23:54:19:a1:95 pdu1:3
```

act_models directory

The *act_models* directory contains files that describe the names of sensor values available via IPMI interfaces.

Using the Act_utils Commands

Act_utils consists of the following commands:

- **act_authsync -** Sychronize user authentication data
- **act_cfgfile -** Generate configuration files
- **act_console -** Connect to a node via serial console or IPMI serial-over-LAN (SOL)
- **act_cp -** Copy files
- **act_dump -** Dumps the configuration in a machine readable format
- **act_exec -** Execute commands
- **act_info -** Gather information usable by tech support
- **act_ipmi_log -** Show or clear the IPMI event log
- **act_ipmi_netcfg -** Used to set the IP addresses on IPMI cards. Also dumps DHCP configuration

for IPMI cards.

- **act_locate** - Turn the chassis locate LED on or off.
- **act_mpi_test** - Used to test bandwidth and latency between nodes
- **act_netboot** - Change network boot options
- **act_nodecompare** - Compares the hardware configuration between nodes
- **act_nodenames** - Prints a list of node names
- **act_powerctl** - Control electrical power
- **act_sensors** - Gather sensor data

Options Common To Most Act_utils Commands

The following options are used for all commands except **act_dump** and **act_cfgfile**:

- **-n** / **--nodes** - A comma separated list of nodes to act upon
- **-g** / **--groups** - A comma separated list of groups to act upon
- **-a** / **--all** - Act upon all nodes
- **-x** / **--exclude** - A comma separated list of nodes to be excluded from the action
- **-r** / **--range** - A range of nodes to act upon. **--nodes=node02-node05** or **--nodes=node02..node05** will act upon node02, node03, node04, and node05
- **-f** / **--file** - Used to specify an alternate configuration file for Act_utils
- **-d** / **--dryrun** - Prints what would be done but does not execute any commands
- **-s / --serial** - Acts on the range of hosts in order.
- **-h** / **--help** - Prints help and usage information

act_authsync

act_authsync is used to synchronize user data amongst the nodes of the cluster. This must be run after adding or deleting users.

Usage

act_authsync ***[OPTIONS]***

Additional Options

act_authsync takes no additional options.

Examples

- Copy the user authentication data to all nodes:

```
$ act_authsync -a
```

- Copy the user authentication the the “nodes” group:

```
$ act_authsync -g nodes
```

act_cfgfile

act_cfgfile is used to create specific configuration files from the data contained in the Act_utils configuration. Any existing files are copied to *filename.1*, up to 5 levels of previous files.

Usage

act_cfgfile ***[OPTIONS]***

Options

- **-h** / **--help** - Print a help and usage message
- **-p** / **--prefix** - Directory to write generated files. The default is */tmp/cluster_cfg*. **act_cfgfile** is most useful with the prefix set to **/**
- **-s** / **--stdout** - Print configuration files to STDOUT instead of writing files
- **--hosts** - Generate */etc/hosts* for all known devices
    - **--host_output** - Specify an alternate destination file
- **--ssh** - Generate *ssh_known_hosts2*
    - **--ssh_output** - Specify an alternate destination file
    - **--ssh_keyfile** - Extract the host key from an alternate file
- **--dhcp** - Generate a file to be included in the configuration for dhcpd. Default: */etc/dhcpd.d/nodes.conf*
    - **--dhcp_output** - Specify an alternate destination file
    - **--dhcp_device** - Specify which NIC to use. Default: eth0
- **--cloner** - Generate node specfic files for cloner. Requires **--cloner_image**, **--cloner_os**, and **--cloner_dev**
    - **--cloner_image** - The image to which the files are to be generated
    - **--cloner_path** - Specify an alternate path to the cloner software
    - **--cloner_dhcp** - Specify that DHCP is to be used instead of static addresses
    - **--cloner_os** - Specify the OS of the image. Currently only **redhat** is supported. **redhat** covers Red Hat Enterprise Linux, Fedora, and all Red Hat clone distributions
- **--torque** - Generate a nodes file for a TORQUE server. Requires **--torque_ppn**
    - **--torque_output** - Specify an alternate destination file
    - **--torque_ppn** - Specifies the cores per node
- **--bind** - Generates DNS zone files. Requires **--bind_server**
    - **--bind_server** - Full hostname of the server that will host the zone.
    - **--bind_path** - Alternate path to output zone files
- **--nagios** - Generates Nagios configuration files.
    - **--nagios_group** - Path to Nagios configuration files.
    - **--nagios_dev** - Path to Nagios device configuration files

Examples

- Generate hosts file:

```
$ act_cfgfile --hosts --prefix=/
```

act_console

act_console is used to connect to a node via serial console, such as OpenGear, or IPMI serial-over-LAN (SOL) interface.

Usage

act_console ***[OPTIONS]***

Additional Options

- **--use_xterm** - Start the console connection in an Xterm windows. Requires X
- **--xterm** - Specifiy an alternate X terminal program such as **gnome-terminal**

Examples

- Start a console connection to node01:

```
$ act_console --nodes=node01
```

- Start console connection in Xterm windows to node01, node02, and node03:

```
$ act_console --range=node01,node03 --use_xterm
```

act_cp

act_cp is used to copy files to the nodes in the cluster.

Usage

act_cp ***[OPTIONS] SOURCE [DESTINATION]***

DESTINATION may be omitted, in which case ***SOURCE*** is used as ***DESTINATION.***

Additional Options

- **--recursive** - Copy files recursively such as copying an entire directory
- **-p** / **--preserve** - Preserve ownership and permissions on copied file. If omitted the file will be copied as the user executing **act_cp**

Examples

- Copy */tmp/src_file* to */tmp/dst_file* on all nodes in the nodes group:

```
$ act_cp -g nodes /tmp/src_file /tmp/dst_file
```

- Copy the directory */tmp/dir* to */tmp/dir* to node01 and node01 recursively and preserve ownership:

```
$ act_cp --nodes=node01,node02 --recursive --preserve /tmp/dir /tmp/dir
```

act_dump

Dumps the data contained in the configuration file in a machine parsable format.

act_exec

act_exec is used to execute commands.

Usage

act_exec ***[OPTIONS] COMMAND***

Normal shell redirection can be used in the COMMAND be enclosing the command in quotes. If double quotes (") are used variable expansion will occur before *act_exec* reads the options. Using single quotes (') will pass any variable on to be expanded on the target node(s).

Additional Options

- **-s** / **--serial** - Run the command in serial on the nodes. This is slower than the default parallel action. Serial return results in the same order each time.

Examples

- Execute **ls /tmp** on all nodes in the nodes group:

```
$ act_exec -g nodes ls /tmp
```

- Run **dmesg** and pipe it through **grep** looking for any lines containing the string "eth0". This is on all nodes:

```
$ act_exec -a 'dmesg | grep eth0'
```

- Echo the contents of the HOSTNAME variable on node01 and node02:

```
$ act_exec --nodes=node01,node02 'echo $HOSTNAME'
```

act_info

act_info is used to gather information about nodes that may be useful to Advanced Clustering's technical support staff.

Usage

act_info ***[OPTIONS] OUTPUT_FILE***

Additional Options

- **--tmp_path** - Directory in which temporary files are to be written

Examples

- Gather data from node01 through node05 and save it in *data_1-5.tgz*:

```
$ act_info --range=node01,node05 data_1-5.tgz
```

act_ipmi_log

View and clean the IPMI event log.

Usage

act_ipmi_log [*OPTIONS*] *COMMAND*

Commands

- **list** - Shows the contents of the IPMI event log.
- **clear** - Clears the contents of the IPMI event log.
- **date** - Shows the current date.

act_ipmi_netcfg

Configure the IPMI LAN interface and the selected hosts. The OpenIPMI drivers must be loaded on the target hosts.

Usage

./act_ipmi_netcfg [*OPTIONS*]

Additional Options

- **--lanchannel=[*NUM*]** - The IPMI channel to set (default: 1)
- **--dump_dhcp** - Dumps a BIND config file with the MAC addresses of the IPMI interfaces.

Examples

Set the IP settings for the IPMI controller in nodes node01, node05

```
act_ipmi_netcfg --nodes=node01,node05
```

Dump a DHCP config file for use with BIND for the IPMI controller in nodes node01, node05

```
act_ipmi_netcfg --nodes=node01,node05 --dump_dhcp
```

act_locate

Used to turn on the locator LED on nodes with locator LEDs.

Usage

act_locate [*OPTIONS*] *COMMAND*

Commands

- **on** - Turns on the locator LED
- **off** - Turns off the locator LED

Examples

- Turn on the locator LED in the group 'rack1'

```
$ act_locate --group=rack1 on
```

act_mpi_test

Used to test node to node communication using MPI. For the nodes selected, each node will test its communication to every other selected node.

Usage

act_mpi_test [*OPTIONS*] *OUTPUT_DIR*

Additional Options

- **--avg_dev** - Percent of deviation from average that is allowed. Values outside of this will be marked in the results.
- **--mpipath** - Path to the preferred MPI implementation.
- **--mpitype** - Types are: mpich1, mpich2, openmpi.
- **--tmpdir** - Temporary directory. Must be on a shared filesystem.

Examples

- Test nodes 1 though 5 using MVAPICH2. Results are placed in the users home directory.

```
$ act_mpi_test --range=node01-node05 --mpipath=/act/mvapich2/gnu
--mpitype=mpich2 --tmpdir=~/tmp --avg_dev=10 ~/testresults/
```

act_netboot

act_netboot is used to adjust the network boot options of the nodes.

Usage

act_netboot *[OPTIONS]*

Additional Options

- **--tftp_root** - The root of the TFTP directory
- **--template_path** - The path to the directory containing the netboot templates
- **--symlink_path** - The relative path to the templates from the TFTP directory
- **--templates** - Prints the list of installed netboot templates
- **--list** - Prints a list of nodes and their selected netboot template
- **--set** - Set the template to be used for the requested nodes
- **--delete** - Delete the template associated with the requested nodes

Examples

- Show the available templates:

```
$ act_netboot --templates
```

- Set all nodes in the *nodes* group to boot using the breakin template:

```
$ act_netboot -g nodes --set=breakin
```

- Set all nodes to boot to their local disk:

```
$ act_netboot --all --set=localboot
```

act_nodecompare

act_nodecompare is used to check the hardware configuration of the nodes.

Usage

act_nodecompare *[OPTIONS] OUTPUT*

Additional Options

- **--format** - Out file format. Can be text or html
- **--tests**- Comma separated list of test to be run
    - bios - System BIOS version
    - corecount - CPU core count
    - diskmodel - Model of hard disk
    - diskperf - Disk read performance with hdparm
    - ethdev - Collect information on ehternet devices

- kernel - Kernel version
- memory - Total amount of system RAM
- memperf - Memory performance from STREAM Triad results
- pcidev - md5sum of the PCI device list.

Examples

- Print a results of all tests on node01-node05 and write to compare.txt

```
$ act_nodecompare --range=node01-node05 compare.txt
```

- Just check the memory performance and disk performance on node01-node05

```
$ act_nodecompare --nodes=node01,node05 --tests=memperf,diskperf compare.txt
```

act_nodenames

act_nodenames is used to print a list of node names.

Usage

act_nodecompare *[OPTIONS]*

Additional Options

- **--delimiter** - The character or string to be printed between eachnode name.

Examples

- Print the names of the nodes in the group “nodes”

```
$ act_nodenames --group=nodes
```

act_powerctl

act_powerctl is used to control electrical power via PDU or IPMI.

Usage

act_powerctl *[OPTIONS] COMMAND*

Additional Options

COMMAND may be one of the following:

- **status** - Print the status of the power
- **on** - Turn on the power
- **off** - Turn off the power

- **reboot** - Cycle the power off then on

Examples

- Turn on all nodes in the group nodes:

```
$ act_powerctl -g nodes on
```

- Reboot node02:

```
$ act_powerctl -n node02 reboot
```

act_sensors

act_sensors gathers data from the sensors on IPMI interfaces. This program can be run as a daemon to display the data in Ganglia. Daemon mode is controlled by */etc/init.d/act_sensors*.

Usage

act_sensors ***[OPTIONS] [COMMAND]***

Additional Options

- **--all_ipmi** - Performs the action on all nodes that support IPMI
- **--csv** - Print retrieved data in a CSV format
- **--gmetric_path** - Path to the gmetric program
- **--gmetric_interval** - Frequency in minutes to update Ganglia

COMMAND may be one of the following:

- **fans** - Display fan speeds
- **gmetric** - Send data to Ganglia
- **gmetric**-server - Start in daemon mode to update Ganglia
- **model-template** - Create a template based on available IPMI data
- **sensors** - Display all sensor values
- **temps** - Display temperature values
- **voltages** - Display voltage values

Examples

- Display temperture data from all IPMI supporting nodes:

```
$ act_sensors --all_ipmi temps
```

act_dump

act_dump prints the configuration data in a machine parseable format. The format used is that of Perl's Data::Dumper module.

Usage

```
act_dump
```

Using IPMI

IPMItool is a command line utility for managing and configuring systems with IPMI support. This tool allows you the ability to communicate directly with the IPMI card installed in the server or over a network using TCP/IP. IPMItool is cross platform, and can be installed on another system for remote querying or management of IPMI enabled systems (many operating systems are supported - including:). All clusters running Linux from Advanced Clustering will come with the IPMItools package pre-installed for you. If you would like to download a copy of the tool for another workstation or system, visit the IPMItool website at: online at http://ipmitool.sourceforge.net/.

Basic IPMItool syntax

Connecting locally

When connected locally use the following command: **ipmitool *[CMD]***

Note: The Linux kernel modules *ipmi_si* and *ipmi_devintf* must be loaded for the local access to work. For more information about setting up the operating system for local IPMI support see the section XXXXXX above.

Connecting to a remote system

When connecting to a remote IPMI enabled system:

```
ipmitool -I lanplus -H [HOST/IP] -U [USERNAME] -P [PASSWORD] [CMD]
```

Note: The remote access settings do not require the operating system be configured for IPMI. The IPMI interface will respond to commands as long as the machine is plugged into power. The system does not have to be powered on, or running on operating system.

Option	Description
-I lanplus	Tells IPMI tool you will be connecting to a remote IPMI card with the LAN+ protocol (which is the correct choice for all Advanced Clustering shipping systems).
-H [HOST/IP]	The hostname or IP address of the remote server's IPMI interface (this is not the same as the operating system assigned IP address)
-U [USERNAME] -P [PASSWORD]	the username and password used to authenticate to IPMI (this is not the same as the operating system username and password)
CMD	The IPMItool command to execute (outlined in the sections below)

In the examples below this options will be referenced as ***[CONNECT OPTIONS]*** either replace them with nothing for a locally executed IPMI command, or with the full string of interface, username, password and hostname.

Network settings

Note: When setting the network configuration options it's advised to do this connected locally to the machine you want to make changes.

View existing network settings

To view the existing network settings of the IPMI interface:

```
$ ipmitool [CONNECT OPTIONS] lan print 1

Set in Progress         : Set Complete
Auth Type Support       : NONE MD2 MD5 OEM
Auth Type Enable        : Callback : NONE MD2 MD5 OEM
                        : User     : NONE MD2 MD5 OEM
                        : Operator : NONE MD2 MD5 OEM
                        : Admin    : NONE MD2 MD5 OEM
                        : OEM      :
IP Address Source       : DHCP Address
IP Address              : 172.20.217.44
Subnet Mask             : 255.255.0.0
MAC Address             : 00:24:8c:20:a8:1c
SNMP Community String   : AMI
IP Header               : TTL=0x00 Flags=0x00 Precedence=0x00 TOS=0x00
BMC ARP Control         : ARP Responses Disabled, Gratuitous ARP Disabled
Gratituous ARP Intrvl   : 0.0 seconds
Default Gateway IP      : 172.20.0.1
Default Gateway MAC     : 00:04:23:ae:f3:21
Backup Gateway IP       : 0.0.0.0
Backup Gateway MAC      : 00:00:00:00:00:00
802.1q VLAN ID          : Disabled
802.1q VLAN Priority    : 0
RMCP+ Cipher Suites     : 1,2,3,6,7,8,11,12,0,0,0,0,0,0,0,0
Cipher Suite Priv Max   : aaaaXXaaaXXaaXX
                        :     X=Cipher Suite Unused
                        :     c=CALLBACK
                        :     u=USER
                        :     o=OPERATOR
                        :     a=ADMIN
                        :     O=OEM
```

Statically assign the IP address

To set the network interface to a static IP address, first set the IP assignment method:

```
$ ipmitool [CONNECT OPTIONS] lan set 1 ipsrc static
```

Next set the IP address, netmask, and gateway

```
$ ipmitool [CONNECT OPTIONS] lan set 1 ipaddr X.X.X.X
$ ipmitool [CONNECT OPTIONS] lan set 1 netmask X.X.X.X
$ ipmitool [CONNECT OPTIONS] lan set 1 defgw ipaddr X.X.X.X
```

Assign the IP address via DHCP

If you'd like to set the IP address statically through DHCP the "lan print 1" command listed above will show you the hardware MAC address to use.

```
$ ipmitool [CONNECT OPTIONS] lan set 1 ipsrc dhcp
```

Username / Password

The IPMI card on each server has it's own username and password database. Each user is identified by a numeric ID and all of the user modification commands are done based on this ID number.

The following sections will outline how to determine user ID numbers, and manage the authentication database.

Listing users

Use the command "user list 1" to view the IPMI card's authentication database:

```
$ ipmitool [CONNECT OPTIONS] user list 1

ID  Name        Callin  Link Auth   IPMI Msg   Channel Priv Limit
1                    false   false      true       ADMINISTRATOR
2   root             false   false      true       ADMINISTRATOR
3   admin            false   false      true       ADMINISTRATOR
```

Adding a new user

First, select pick an unused ID (see list above) and set the username:

```
$ ipmitool [CONNECT OPTIONS] user set name [ID] [USERNAME]
```

Next, set the users password using the same ID as above:

```
$ ipmitool [CONNECT OPTIONS] user set password [ID] [PASSWORD]
```

Finally, set the use's privilege level, using one of the following privilege levels:

Privilege Level	Description
3	Operator: can manage the server but not add or delete new users
4	Administrator: has full IPMI privileges

```
$ ipmitool [CONNECT OPTIONS] user priv [ID] [LEVEL]
```

Example that creates a new administrator user named "myuser" with the password "mypass":

```
$ ipmitool [CONNECT OPTIONS] user list 1

ID  Name        Callin  Link Auth   IPMI Msg   Channel Priv Limit
1                    false   false      true       ADMINISTRATOR
2   root             false   false      true       ADMINISTRATOR
3   admin            false   false      true       ADMINISTRATOR

$ ipmitool [CONNECT OPTIONS] user set name 4 myuser
```

```
$ ipmitool [CONNECT OPTIONS] user set password 4 mypass
$ ipmitool [CONNECT OPTIONS] user priv 4 4 1
$ ipmitool [CONNECT OPTIONS] user list

ID  Name        Callin  Link Auth   IPMI Msg   Channel Priv Limit
1                    false   false      true       ADMINISTRATOR
2   root             false   false      true       ADMINISTRATOR
3   admin            false   false      true       ADMINISTRATOR
4   myuser           true    false      true       ADMINISTRATOR
```

Changing a user's password

Use the command "user set password" to change a user's password, replacing [ID] with the user's numeric ID and [PASSWORD] with the new password.

```
$ ipmitool [CONNECT OPTIONS] user set password [ID] [PASSWORD]
```

Example, change the password for the user named "myuser" to "abc123":

```
$ ipmitool [CONNECT OPTIONS] user list

ID  Name        Callin  Link Auth   IPMI Msg   Channel Priv Limit
1                    false   false      true       ADMINISTRATOR
2   root             false   false      true       ADMINISTRATOR
3   admin            false   false      true       ADMINISTRATOR
4   myuser           true    false      true       ADMINISTRATOR

$ ipmitool [CONNECT OPTIONS] user set password 4 abc123
```

Changing user's privilege level

Use the command "user priv" to change a user's privilege level, replacing [ID] with the user's numeric ID and [LEVEL] with the privilege level:

Privilege Level	Description
3	Operator: can manage the server but not add or delete new users
4	Administrator: has full IPMI privileges

```
$ ipmitool [CONNECT OPTIONS] user priv [ID] [LEVEL] 1
```

Querying sensors

The "sdr" command is used for reading the sensor data repository. You can list all sensors or just get sensors of a particular type. Valid sensor types supported on Advanced Clustering systems include: Temperature, Voltage, and Fan.

To query all sensors:

```
$ ipmitool [CONNECT OPTIONS] sdr

CPU1 Temperature | 40 degrees C      | ok
CPU2 Temperature | 33 degrees C      | ok
TR1 Temperature  | 0 degrees C       | cr
TR2 Temperature  | 0 degrees C       | cr
```

```
VCORE1           | 1.06 Volts        | ok
VCORE2           | 0.94 Volts        | ok
+1.5V_ICH        | 1.54 Volts        | ok
+1.1V_IOH        | 1.10 Volts        | ok
......
```

To view sensors of a particular type:

```
$ ipmitool [CONNECT OPTIONS] sdr type [TYPE]
```

Example, get all voltages:

```
$ ipmitool [CONNECT OPTIONS] sdr type Voltage

VCORE1           | 34h | ok  |  0.0 | 1.06 Volts
VCORE2           | 35h | ok  |  0.0 | 0.94 Volts
+1.5V_ICH        | 39h | ok  |  0.0 | 1.54 Volts
+1.1V_IOH        | 3Ah | ok  |  0.0 | 1.10 Volts
+3.3VSB          | 40h | ok  |  0.0 | 3.24 Volts
+3.3V            | 36h | ok  |  0.0 | 3.19 Volts
+12V              | 38h | ok  |  0.0 | 12.10 Volts
VBAT             | 3Ch | ok  |  0.0 | 3.14 Volts
+5VSB            | 3Bh | ok  |  0.0 | 5.25 Volts
+5V              | 37h | ok  |  0.0 | 5.09 Volts
P1VTT            | 3Dh | ok  |  0.0 | 1.14 Volts
P2VTT            | 3Fh | ok  |  0.0 | 1.13 Volts
+1.5V_P1DDR3     | 3Eh | ok  |  0.0 | 1.52 Volts
+1.5V_P2DDR3     | 41h | ok  |  0.0 | 1.53 Volts
```

Power control

IPMItool has complete power control over the host system. This includes power on, off, and reset. Note: You should perform power control over the network and not locally.

Check the power status

Use the "chassis power status" command to query the current state of the system.

```
$ ipmitool  [CONNECT OPTIONS] chassis power status
```

Chassis Power is on

Power off the system

Use the "chassis power off" command to turn off the power of the system. This is hard power off - the operating system will not be shutdown properly.

```
$ ipmitool [CONNECT OPTIONS] chassis power off
```

Power off the system (clean shutdown)

Use the "chassis power soft" command to initiate a soft-shutdown of the OS via ACPI. In Linux this will trigger a "shutdown" command and safely shutdown the operating system, and then power off the system.

```
$ ipmitool [CONNECT OPTIONS] chassis power soft
```

Power on the system

Use the "chassis power on" command to power on the system.

```
$ ipmitool [CONNECT OPTIONS] chassis power on
```

Reset a system

Use the "chassis power reset" command to do a hard reset on the system. This is hard reset - the operating system will not be shutdown properly.

```
$ ipmitool [CONNECT OPTIONS] chassis power reset
```

Connecting to the text console

IPMI supports redirecting serial console output over the network. This allows BIOS level access, and the ability to watch a system boot, and monitor any error output sent to the console.

Use the "sol activate" command to connect to a remote text console. Note: This only works through the lan interface and not locally.

```
$ ipmitool [CONNECT OPTIONS] sol activate

[SOL Session operational.  Use ~? for help]
```

Once connected you will have to use the hot key "~." to disconnect from the serial over lan connection.

Event log viewer

The system contains an event log that has can contain some useful information. If you'd like to browse the event log you can use the "sel elist" command:

```
$ ipmitool [CONNECT OPTIONS] sel elist
```

Using Cloner

Features

Cloner allows a system administrator to replicate the operating system and configuration to the nodes in a cluster. Multicast is supported to allow many nodes to be installed at one time. Cloner uses the concept of *images*, a snapshot of an existing node. One node is installed and configured, referred to as a *golden node*, and an image is created.

There are two main components to Cloner: an image collection program named *cloner* and a small installer environment that is loaded from a PXE/TFTP server, CD-ROM or even a USB key. *cloner* is run on the *golden node*, or the node that will be used as the basis of the image. The image is uploaded to the Cloner server, usually the head node of the cluster. Destination hosts load the installer environment and copy the contents of the image to their local storage device.

Usage

Creating An Image With The cloner Command

The *cloner* command copies the data from the installed node, the one that _cloner_is being run on, and copies it to the Cloner server.

The following command line parameters are accepted:

- **`--server=`*`hostname`*** - the IP address or hostname of the Cloner server, usually the head node.
- **`--image=`*`name`*** - the title of the image to be created.
- **`--onself`** - this parameter is used when taking an image of a Cloner server. This prevents the server from attempting to copy the image that it is currently creating.
- **`--update`** - updates an existing image with the current contents of the node. This is most often used when software needs to be added to the nodes in a cluster.
- **`--dryrun`** - displays what would be done but does not create or update the image.
- **`--nosync`** - only copies the setup information not the filesystem contents.
- **`--ignorelvm`** - do not try to detect LVMs.
- **`--nolastlog`** - Do not sync /var/log/lastlog

Creating an Image of a Node
The following examples treat are run on the *golden node*, the source of the image.

Create an image named ***`node_image`*** on the server named ***`head`***.

```
$ /act/cloner/bin/cloner --server=head --image=node_image
```

Updating an Existing Node Image

When changes are made on the *golden node*, the *--update* parameter is used to copy the changes to the image. In this case we will update *node_image* with the current data on the *golden node*.

```
$ /act/cloner/bin/cloner --server=head --image=node_image --update
```

Installing An Image On A Node

The nodes in an Apex Cluster are pre-configured to boot via PXE/TFTP. A screen with a blue background will be presented; it's title will be *Advanced Clustering PXE Menu*. Use the arrow keys to highlight the entry titled *Cloner* and press the *TAB* key on the keyboard to edit the boot command line. The Cloner installer accepts the following additional options

- **server=*ip address*** - the IP address of the Cloner server, usually the head node.
- **image=*name*** - name of the cloner image that is to be installed on the target node.
- **srcpath=*CD/DVD-ROM device*** - the path to the CD or DVD-ROM device if installing from a CD or DVD.
- **manualdisk=*1*** - does not partition the disks, the person installing the image will have to manually create and mount the filesystems.
- **multicast=*1*** - perform a multicast install allowing many nodes to be installed at one time.
- **_netmod=*module1,module2*** - network card modules to be loaded in the given order.
- **node=*node name*** - the name of the target node.

To install node02 with the image name ***node_image*** add the following after pressing **TAB**:

```
server=10.1.1.254 image=node_image node=node02
```

Installing An Image On Multiple Nodes at Once (Multicast)

On the same blue network PXE boot menu, highlight the entry titled *Cloner Multicast* and press the *TAB* key on the keyboard to edit the boot command line. The Cloner Multicast installer accepts the same options as the standard Cloner installer.

To install your nodes with the image name *node_image*

- On the headnode, run:

```
$ /act/cloner/bin/cloner_mcastsrv --image=node_image --interface=eth0
```

- Boot each node selecting “Cloner Multicast”, and add the following after pressing **TAB**:

```
$ server=10.1.1.254 multicast=1 image=node_image node=nodename
```

- After all nodes are booted and have entered “Cloner Multicast”, press enter on the head node.

Adding Nodes With Act_utils

Act_utils greatly simplifies the process of adding new nodes.

The examples given in this section assume the following:

- The cluster has 1 head and 2 compute nodes: head, node01, node02

- Two nodes will be added: node03, node04
- The new nodes will be in the same Act_utils group as the existing compute nodes. The group is named nodes

Required Information

The following information will be needed to add new nodes:

- Names of the new nodes
- IP Addresses to be assigned to the new nodes
- MAC Addresses of the new nodes

If the new nodes were purchased from Advanced Clustering, also note the serial number of the nodes. If the nodes do not have an Advanced Clustering serial number create a unique number for each node.

Adding to act_nodes.conf

/act/etc/act_nodes.conf defines the node names, serial numbers and MAC addresses. See page 30 for complete information on *act_nodes.conf*.

The information for the 2 new nodes is presented in bolded text:

```
head   325128 eth0=00:15:17:26:87:de,eth1=00:15:17:26:87:dc pdu1:1,pdu2:1 console1:48
node01 399999 eth0=00:1f:c6:0c:d4:0a,eth1=00:1f:c6:0c:d4:0b none          console1:1
node02 324757 eth0=00:23:54:19:a1:43,eth1=00:23:54:19:a1:95 pdu1:3
node03 111111 eth0=00:00:00:00:00:00,eth1=00,00,00,00,00,01
node04 111112 eth0=00:00:00:00:00:02,eth1=00,00,00,00,00,03
```

Adding to act_util.conf

/act/etc/act_util.conf defines the IP addresses and groups of nodes. See page 26 for complet information on *act_nodes.conf*.

Below is the current definition of the “nodes” group. This defines the IP addresses for node01 and node02:

```
[nodes]
type=range
hostname_prefix=node
hostname_start=1
hostname_end=2
ip_start=10.1.1.1
ip_end=10.1.1.2
pad=2
```

To add node03 and node04 hostname_end and ip_end will be modified:

```
[nodes]
type=range
hostname_prefix=node
hostname_start=1
hostname_end=4
ip_start=10.1.1.1
ip_end=10.1.1.4
pad=2
```

Generating Configuration Files

Common configuration files are created/updated with the **act_cfgfile** command (page 32). The following command will add the new nodes to the *hosts* files , add the nodes to DHCP, create SSH key entries and set them to use the cloner image named “node”.

```
$ act_cfgfile --prefix=/ --hosts --dhcp --dhcp_device=eth0 --ssh --cloner \
    --cloner_image=node --cloner_dev=eth0
```

/etc/hosts and /etc/ssh/ssh_known_hosts should be copied to the existing nodes:

```
$ act_cp -g nodes /etc/hosts /etc/
$ act_cp -g nodes /etc/ssh/ssh_known_hosts2 /etc/ssh/
```

Update the Cloner Image

Login to node01, the “golden node” (see page 47), and update the cloner image:

```
$ ssh node01
$ /act/cloner/bin/cloner --server=head --image=node --update
$ logout
```

Change the Network Boot Settings to Cloner

Using act_netboot, set the new nodes to boot to cloner:

```
$ act_netboot --nodes=node03,node04 --set=cloner-v2.3
```

Network Boot the New Nodes

Nodes from Advanced Clustering will be set to automatically network boot. Simply power them on with their network cables plugged in.

Cloner will display it's status on the screen of the node being installed.

Change the Network Boot Settings to the Local Disk

Once cloned, the nodes will be set to boot from their local disks:

```
$ act_netboot --nodes=node03,node04 --set=localboot
```

Reset The New Nodes

Once cloner has finished, reset the cloned nodes and they will boot to their OS on their local disk.

Appendices

Key files in the ''images'' directory

File	Purpose
modules	a list of modules to load on startup (primary use for SCSI controllers)
filesystems	format "device mntpoint fstype extlabel"
bootloader	format "device [grub|lilo]"
lvm.VOLNAME	the config file to restore lvm's (one file per each volume group)
makedirectories	directories to create on the filesystem, format "path mode uid gid"
pv_devices	the LVM physical volumes, format "device volgrp UUID"
DEV.sfdisk	file used to re-partition the disk (piped through sfdisk -uM)

Running Parallel Programs

Advanced Clustering simplifies the process of running a parallel program by including a queuing system (TORQUE), example submission scripts, and the *module* command that allows users to easily switch their parallel library environment.

Introduction To Queuing

Queuing systems monitor and allocate the nodes of a cluster to users. Running all jobs through a queuing system prevents multiple users from running on the same node causing poor performance or even causing the node to crash by exhausting the memory of the node. Users can also submit many jobs at one time and they will be run as the resources are available.

TORQUE provides many features that are not addressed in this manual. Cluster Resources has more thorough documentation at http://www.clusterresources.com/products/torque/docs/

Creating An Environment

When a user first logs in to an Apex Cluster they will be guided through creating a proper environment for compiling and running a parallel program. If a users declines the creation of any of the steps it will be logged in the system log.

In the event that the setup needs to be run again the user can run *touch ~/.actrun* then log out and back in.

The setup creates an SSH key, *~/.mpd.conf*, adds */act/bin* the the *PATH*, and adds the **`module`** command to the user's environment.

Using The module Command

The **`module`** command allows a user to change their environment, including execution paths, library paths, and manual paths. Each Apex cluster has *module* environments created for each MPI implementation included. At the bare minimum these include ***`mpich2/gnu`*** and ***`openmpi/gnu`***. To obtain a complete list run **`module avail`**.

To compile and run a parallel program using MPICH the ***`mpich/gnu`*** module will be loaded.

```
$ module load mpich/gnu
```

The currently loaded modules can be displayed.

```
$ module list
Currently Loaded Modulefiles:
  1) mpich/gnu
```

The multiple MPI versions provide similar commands to ensure that the desired version is used only one MPI module can be loaded at any time. Trying to load another one will result in an error.

```
$ module load mpich2/pgi
mpich2/pgi(5):ERROR:150: Module 'mpich2/pgi' conflicts with the currently loaded
module(s) 'mpich/gnu'
mpich2/pgi(5):ERROR:102: Tcl command execution failed: conflict mpich2 mpich mvapich
mvapich2 openmpi
```

Modules can be unloaded.

```
$ module unload mpich/gnu
```

All modules can be removed.

```
$ module purge
```

Compiling A Simple MPI Program

This is an example session where an MPI program is compiled. *cpi.c*, as included with MPICH2, will be use.

```
$ cd ~/
$ cp /act/mpich2/gnu/share/examples_logging/cpi.c ~/cpi.c
$ module load mpich/gnu
$ mpicc -o cpi cpi.c
```

There will now be an executable program named **cpi**.

Integrating MPI And TORQUE

Apex Clusters include the **mpiexec-act** parallel job launcher. This command integrates MPI and TORQUE simplifying job submission. Example TORQUE submission scripts are available in the directory */act/examples/torque_submission_scripts*. These can be copied to a users home directory and used as a template for their own job submission scripts.

An example submission script using MPICH to run the command **cpi**. This file will be named *cpi.sub*.

```
#!/bin/sh
#
# This script show how to run an MPICH program using mpiexec-act (OSU
# mpiexec). Mpiexec-act uses the requested PBS resources to run on the correct
# nodes.

# Request 2 nodes with 8 cores on each node.
#PBS -l nodes=2:ppn=2
#
# Set the output and error to one file and name that file CPI_OUTPUT
#PBS -j oe
#PBS -o CPI_OUTPUT

# Load envrionment modules
if [ -f /act/Modules/3.2.6/init/bash ]; then
        source /act/Modules/3.2.6/init/bash
else
        echo "Could not source environment modules!"
        exit 1
fi

# Unload any modules and load the mpich/gnu, mpich/pgi or mpich/intel module.
module purge
module load mpich/gnu
```

```
# Use mpiexec-act to launch the program cpi-mpich2. This uses the PMI interface
# to launch the processes.
cd ~/
mpiexec-act -comm p4 ./cpi
```

Examples are provided for the major MPI versions: MPICH, MPICH2, and Open MPI. On InfiniBand equipped clusters, MVAPICH and MPICH are analogous as are MVAPICH2 and MPICH2.

Open MPI does not use **`mpiexec-act`** as it's **`mpiexec`** command provides similar features.

Submitting A Job

Once a submission script is created it is submitted to the queue with the **`qsub`** command.

```
qsub cpi.sub
83.head.cluster
```

In the above example the job id is 83 and it was submitted to head.cluster.

Checking The Status Of A Queue

The command **`qstat -a`** will show the jobs in a queue. This includes running jobs and jobs that have not yet started.

Deleting A Job

Jobs are removed from the queue with the **`qdel`** command. To delete the job with the ID of 83:

```
$ qdel 83
```

In many cases MPI jobs will not exit completely. These processes will need to be killed on each node. Killing hung processes can be made easier with the act_exec command, see page 26 for more information.

InfiniBand

InfiniBand is a high speed interconnect that is available on Apex Clusters. Parallel programs with lots of communication benefit greatly from the low latency and high bandwidth provided by InfiniBand.

Many users are unfamiliar with InfiniBand networks; this section will cover the basic operation of an InfiniBand network.

Subnet Manager

An InfiniBand network requires a *subnet manager*. Apex Clusters use a software based subnet manager named OpenSM. Only one subnet manager may be running. If the head node has an InfiniBand card the subnet manager will run on the head node. If the head node does not have an InfiniBand card the lowest numbered node will be used.

OpenSM is controlled by the init script */etc/init.d/opensmd*. This script accepts the standard ***start***, ***stop***, ***status***, and ***restart*** arguments.

Checking Connectivity

Each InfiniBand port, be it on a switch or a compute node, will have two LEDS. The green LED will illuminate when a physical link is detected, meaning the cable is plugged in at each end. The orange LED will light up once a subnet manager is running and the InfiniBand network is active.

The command **ibstat** is used to view the status of the InfiniBand ports. The following example displays a two port InfiniBand card with one active port.

```
CA 'mthca0'
   CA type: MT25208
   Number of ports: 2
   Firmware version: 5.1.400
   Hardware version: a0
   Node GUID: 0x0002c9020020180c
   System image GUID: 0x0002c9020020180f
   Port 1:
      State: Active
      Physical state: LinkUp
      Rate: 10
      Base lid: 1
      LMC: 0
      SM lid: 1
      Capability mask: 0x02510a6a
      Port GUID: 0x0002c9020020180d
   Port 2:
      State: Down
      Physical state: Polling
      Rate: 10
      Base lid: 0
      LMC: 0
```

```
    SM lid: 0
    Capability mask: 0x02510a68
    Port GUID: 0x0002c9020020180e
```

The “Physical state” field denotes the physical, cable, connection status.

Physical State	Meaning
Polling	The cable is not connected
LinkUp	A physical cable connection has been made

The *State* field indicates the state of the port including the presence of a subnet manager on the network.

State	Meaning
DOWN	There is no cable connected
INIT	A cable is connected but there is no subnet manager on the network
ACTIVE	A cable and subnet manager are both present and the port is ready to be used

Obtaining Support

All of the technicians at Advanced Clustering are well versed in InfinBand. In the event that support for InfiniBand is required please have the following information available for our support technicians:

- The output of **`ibstat`** for all involved nodes
- The output of **`ibcheckerrors`**

See page 7 for contact information.

Warranty

Advanced Clustering Warranty and RMA Procedures

Please read the following material as it has important warranty and RMA procedures.

The following information will be needed when contacting Advanced Clustering:

- Serial number of node(s)
- Site name
- Site address
- Site contact name
- Site contact telephone number
- Site contact email address
- Brief description of the problem
- Steps taken to correct the problem so far

The serial number is important to verify the terms of coverage, as well as determine the correct replacement components and procedures for your system. Our help desk will need to perform a minimum amount of troubleshooting and diagnostics to provide indicators of the appropriate actions required to resolve the issue and which spare parts may be required. If at any time you would like to upgrade your existing coverage, or add additional units, please feel free to contact us by calling toll free at 1.866.802.8222 or e-mailing us at sales@advancedclustering.com.

What to do when you have a service issue:

- Visit our knowledge base at http://www.advancedclustering.com/knowledgebase for answers to the most common questions.
- You may file an on line support ticket at http://www.advancedclustering.com/support
- You may also email support@advancedclustering.com and a ticket will be assigned to you.
- You may call our technical support line at 1.866.802.8222, extension 2. Our representatives are available from 9AM to 5PM, Central Time, Monday to Friday, excluding holidays. If all representatives are busy helping other customers, you may leave a voice message and we will return your call immediately.
- Users with 24x7 support will be prompted for information and their call returned within the period specified in their contract.

Warranty Coverage and Limitations

The Warranty Service Plan provides for the replacement of Server/Workstation components that fail due to manufacturing defects in materials and workmanship. Excluded from warranty coverage are acts of nature, such as electrical storms, floods, fire, etc., acts of war and terrorism, criminal acts, and customer damage and negligence.

Upon receipt of a service request from you, our help desk will contact you promptly to begin diagnosis and arrange a time for the delivery of service. After diagnosis has been performed, the help desk may then issue an RMA for the replacement of parts. You are responsible for shipping the defective parts to Advanced Clustering as instructed during the RMA process. You agree to properly package parts for return and deliver the package(s) to the selected courier for return to the appropriate address. Please contact Advanced Clustering at 1.866.802.8222, extension 2 immediately for assistance with returning parts.

You are responsible for the replacement price of parts that are not returned, and for those damaged in transit due to improper packaging. If parts return shipments are not received before the RMA expires, you will be invoiced for the parts replacement price and an administration fee of 10% of the parts replacement price and no less than $25. The parts delivered to your site at the initiation of service become your property after the returned parts are received, unencumbered by any lien.