Base Users Guide

5.4.3 Edition

Base Users Guide:

5.4.3 Edition

Published Aug 11 2011

Table of Contents

List of Tables

Preface

Since May 2000, the Rocks group has been addressing the difficulties of deploying manageable clusters. We have been driven by one goal: *make clusters easy*. By *easy* we mean easy to deploy, manage, upgrade and scale. We are driven by this goal to help deliver the computational power of clusters to a wide range of scientific users. It is clear that making stable and manageable parallel computing platforms available to a wide range of scientists will aid immensely in improving the state of the art in parallel tools.

Chapter 1. Overview

Table 1-1. Summary

Name	base
Version	5.4.3
Maintained By	Rocks Group
Architecture	i386, x86_64
Compatible with Rocks®	5.4.3

The base roll has the following requirements of other rolls. Compatability with all known rolls is assured, and all known conflicts are listed. There is no assurance of compatiblity with third-party rolls.

Table 1-2. Compatibility

Requires	Conflicts
Kernel	
OS	
Service Pack	

This roll has been released independent of the corresponding Rocks® release. It therefore requires the complete **OS** roll and will not function correctly if using only the **Jumbo** or incomplete set of **OS** CDROMs.

Chapter 2. Attributes

Table 2-1. Roll Attributes

Name	Type	Default
disableServices	string	kudzu canna cWnn FreeWnn kWnn tWnn mDNSResponder
Info_CertificateCountry a	string	
Info_CertificateLocality a	string	
Info_CertificateOrganization a	string	
Info_CertificateState a	string	
Info_CertificateContact a	string	
Info_CertificateLatLong a	string	
Info_CertificateName a	string	
Info_CertificateURL a	string	
Kickstart_DistroDir a	string	/export/rocks
Kickstart_Keyboard a	string	us
Kickstart_Lang a	string	en_US
Kickstart_Langsupport a	string	en_US
Kickstart_Mutlicast a	string	226.117.172.185
Kickstart_PrivateAddress a	string	10.1.1.1
Kickstart_PrivateBroadcast a	string	10.1.255.255
Kickstart_PrivateDNSDomain a	string	local
Kickstart_PrivateDNSServers a	string	10.1.1.1
Kickstart_PrivateGateway a	string	10.1.1.1
Kickstart_PrivateHostname a	string	
Kickstart_PrivateKickstartBaseDir a	string	install
Kickstart_PrivateKickstartCGI a	string	sbin/kickstart.cgi
Kickstart_PrivateKickstartHost a	string	10.1.1.1
Kickstart_PrivateNTPHost a	string	10.1.1.1
Kickstart_PrivateNetmask a	string	255.255.0.0
Kickstart_PrivateNetmaskCIDR a	string	16
Kickstart_PrivateNetwork a	string	10.1.0.0
Kickstart_PrivatePortableRootPassword a	string	
Kickstart_PrivateRootPassword a	string	
Kickstart_PrivateSHARootPassword a	string	
Kickstart_PrivateSyslogHost a	string	10.1.1.1
Kickstart_PublicAddress a	string	
Kickstart_PublicBroadcast a	string	

Name	Type	Default
`Kickstart_PublicDNSDomain` a	string	
`Kickstart_PublicDNSServers` a	string	
`Kickstart_PublicGateway` a	string	
`Kickstart_PublicHostname` a	string	
`Kickstart_PublicKickstartHost` a	string	
`Kickstart_PublicNTPHost` a	string	
`Kickstart_PublicNetmask` a	string	
`Kickstart_PublicNetmaskCIDR` a	string	
`Kickstart_PublicNetwork` a	string	
`Kickstart_Timezone` a	string	
`airboss` b	string	specified on boot line
`arch` c, b	string	i386 | x86_64
`dhcp_filename` d	string	pxelinux.0
`dhcp_nextserver` d	string	10.1.1.1
`hostname` e, b	string	
`kickstartable` d	bool	`TRUE`
`os` c, b	string	linux | solaris
`rack` e, b	int	
`rank` e, b	int	
`rocks_version` a	string	5.4.3
`rsh` f	bool	`FALSE`
`ssh_use_dns` a	bool	`TRUE`
`x11` f	bool	`FALSE`

Notes:
a. Default value created using **rocks add attr `name value`** and affects all hosts.
b. Default value created using **rocks add host attr localhost `name value`** and only affects the `frontend` appliance.
c. Attribute is for internal use only, and should not be altered by the user. Each time a machine installs this attributed is reset to the default value for that machine (depend on kernel booted).
d. Default value created using **rocks add appliance attr `appliance name value`** for the `frontend` and `compute` appliances.
e. Attribute cannot by modified. This value is not recorded in the cluster database and is only available as an XML entity during installation.
f. Attribute is referenced but not defined so is treated as `FALSE`.

`Info_Certificate_{*}`

The attributes are created during `frontend` installation. The values are taken from user input on the system installation screens.

`Kickstart_{*}`

The attributes are created during `frontend` installation. The values are taken from user input on the system installation screens. All of these attributes are considered internal to Rocks® and should not be modified directly.

`airboss`

Specifies the address of the airboss host. This only applies to virtual machines.

`arch`

The CPU architecture of the host. This host-specific attribute is set by the installing machine. User changes to this attribute have no affect.

`dhcp_filename`

Name of the PXE file retrieved over TFTP at startup.

`dhcp_nextserver`

IP address of the server that servers installation profiles (kickstart, jumpstart). In almost all configuration this should be the `frontend` machine.

`kickstartable`

The attribute must be set to `TRUE` for all appliances, and `FALSE` (or undefined) for all unmanaged devices (e.g. network switches).

`os`

The OS of the host. This host-specific attribute is set by the installing machine. User changes to this attribute have no affect.

`rsh`

If `TRUE` the machine is configured as an RSH client. This is not recommended, and will still require RSH server configuration on the `frontend` machine.

`ssh_use_dns`

Set to `FALSE` to disable DNS lookups when connecting to nodes in the cluster over SSH. If establishing an ssh connect is slow the cause may be a faulty (or absent) DNS system. Disabling this lookup will speed up connection establishment, but lowers the security of your system.

`x11`

If `TRUE` X11 is configured and the default runlevel is changed from 3 to 5. X11 is always configure on the `frontend` and this attribute applies only to the other nodes in the cluster.

Chapter 3. Installing a Rocks Cluster

3.1. Getting Started

This chapter describes the steps to build your cluster and install its software.

3.1.1. Supported Hardware

Processors

- x86 (ia32, AMD Athlon, etc.)
- x86_64 (AMD Opteron and EM64T)

Networks

- Ethernet

Specialized networks and components (e.g., Myrinet, Infiniband, nVidia GPU) are also supported. Hardware requirements and software (Rocks Rolls) can be found on the respective vendor web sites.

3.1.2. Minimum Hardware Requirements

Frontend Node

- **Disk Capacity:** 30 GB
- **Memory Capacity:** 1 GB
- **Ethernet:** 2 physical ports (e.g., "eth0" and "eth1")
- **BIOS Boot Order:** CD, Hard Disk

Compute Node

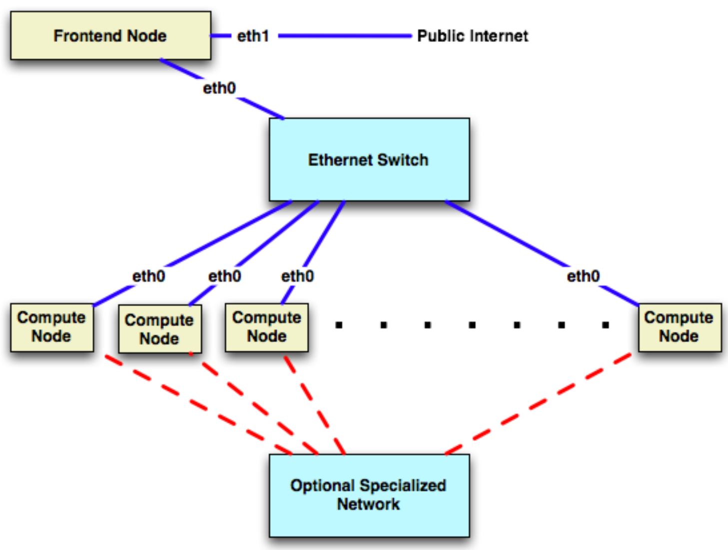
- **Disk Capacity:** 30 GB
- **Memory Capacity:** 1 GB
- **Ethernet:** 1 physical port (e.g., "eth0")
- **BIOS Boot Order:** CD, PXE (Network Boot), Hard Disk

3.1.3. Physical Assembly

The first thing to manage is the physical deployment of a cluster. Much research exists on the topic of how to physically construct a cluster. A majority of the O'Reilly Book[1] *Building Linux Clusters* is devoted to the physical setup of a cluster, how to choose a motherboard, etc. Finally, the book *How to Build a Beowulf* also has some good tips on physical construction.

We favor rack-mounted equipment because of its relative reliability and density. There are Rocks clusters, however, that are built from mini-towers. Choose what makes sense for you.

The following diagram shows how the frontend and compute nodes must be connected:



On the compute nodes, the Ethernet interface that Linux maps to *eth0* should be connected to the cluster's Ethernet switch. This network is considered *private*, that is, all traffic on this network is physically separated from the external public network (e.g., the internet).

On the frontend, at least two ethernet interfaces are required. The interface that Linux maps to *eth0* should be connected to the same ethernet network as the compute nodes. The interface that Linux maps to *eth1* should be connected to the external network (e.g., the internet or your organization's intranet).

3.2. Install and Configure Your Frontend

This section describes how to install your Rocks cluster frontend.

The minimum requirement to bring up a frontend is to have the following rolls:

- Kernel/Boot Roll CD
- Base Roll CD
- Web Server Roll CD

- OS Roll CD - Disk 1
- OS Roll CD - Disk 2

The Core Meta Roll CD can be substituted for the individual Base and Web-Server Rolls.

Additionally, the official Red Hat Enterprise Linux 5 update 4 CDs can be substituted for the OS Rolls. Also, any *true* rebuild of RHEL 5 update 4 can be used -- distributions known to work are: CentOS 5 update 4 and Scientific Linux 5 update 4. If you substitute the OS Rolls with one of the above distributions, you must supply *all* the CDs from the distribution (which usually is 6 or 7 CDs).

1. Insert the Kernel/Boot Roll CD into your frontend machine and reset the frontend machine.

   For the remainder of this section, we'll use the example of installing a *bare-bones* frontend, that is, we'll be using the Kernel/Boot Roll, Core Roll, OS - Disk 1 Roll and the OS - Disk 2 Roll.

2. After the frontend boots off the CD, you will see:

   When you see the screen above, type:

```
build
```

   The "`boot:`" prompt arrives and departs the screen quickly. It is easy to miss. If you do miss it, the node will assume it is a *compute* appliance, and the frontend installation will fail and you will have to restart the installation (by rebooting the node).

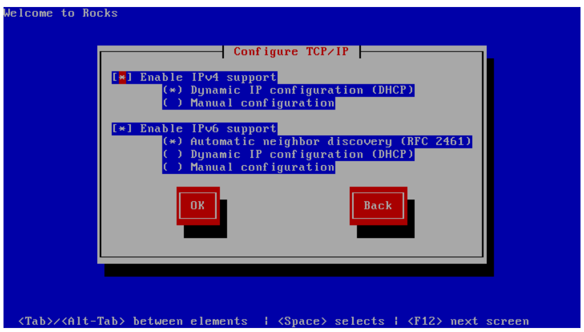
If the installation fails, very often you will see a screen that complains of a missing `/tmp/ks.cfg` kickstart file. To get more information about the failure, access the kickstart and system log by pressing `Ctrl-Alt-F3` and `Ctrl-Alt-F4` respectively.

After you type `build`, the installer will start running.

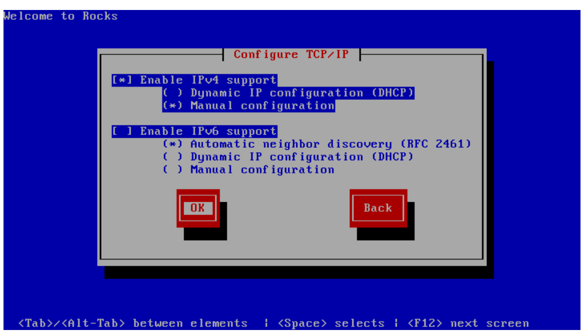
3.

All screens in this step may not appear during your installation. You will only see these screens if there is not a DHCP server on your public network that answers the frontend's DHCP request.

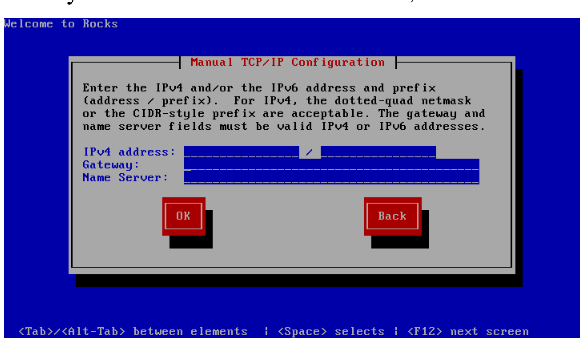
If you see the screen below:



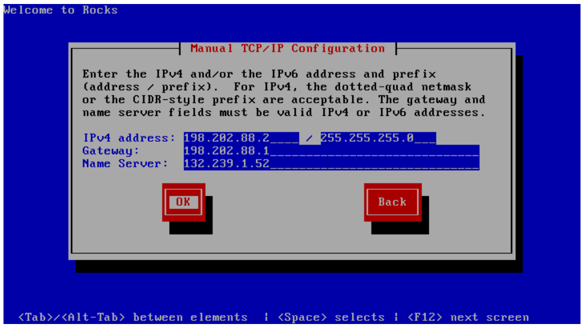
You'll want to: 1) enable IPv4 support, 2) select manual configuration for the IPv4 support (no DHCP) and, 3) disable IPv6 support. The screen should look like:



After your screen looks like the above, hit "OK". Then you'll see the "Manual TCP/IP Configuration" screen:

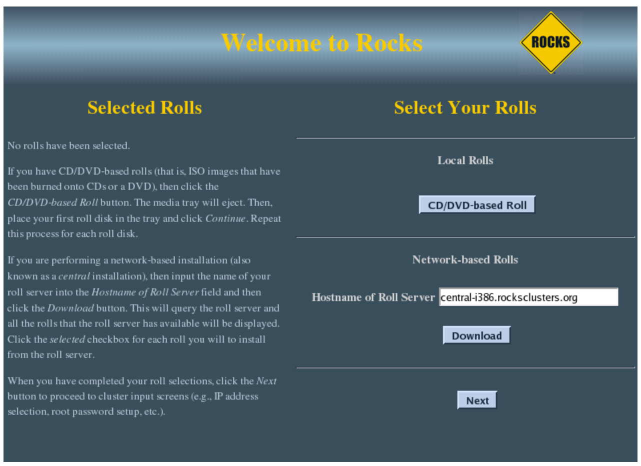


In this screen, enter the *public* IP configuration. Here's an example of the public IP info we entered for one our frontends:



After you fill in the public IP info, hit "OK".

4. Soon, you'll see a screen that looks like:

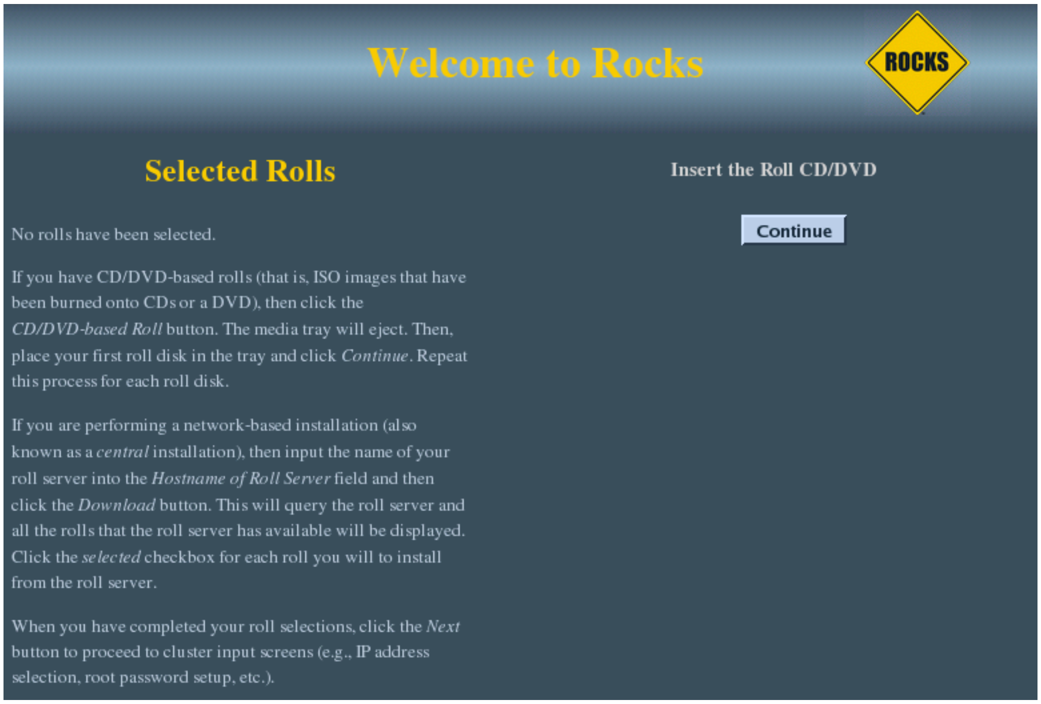


From this screen, you'll select your rolls.

In this procedure, we'll only be using CD media, so we'll only be clicking on the 'CD/DVD-based Roll' button.

Click the 'CD/DVD-based Roll' button.

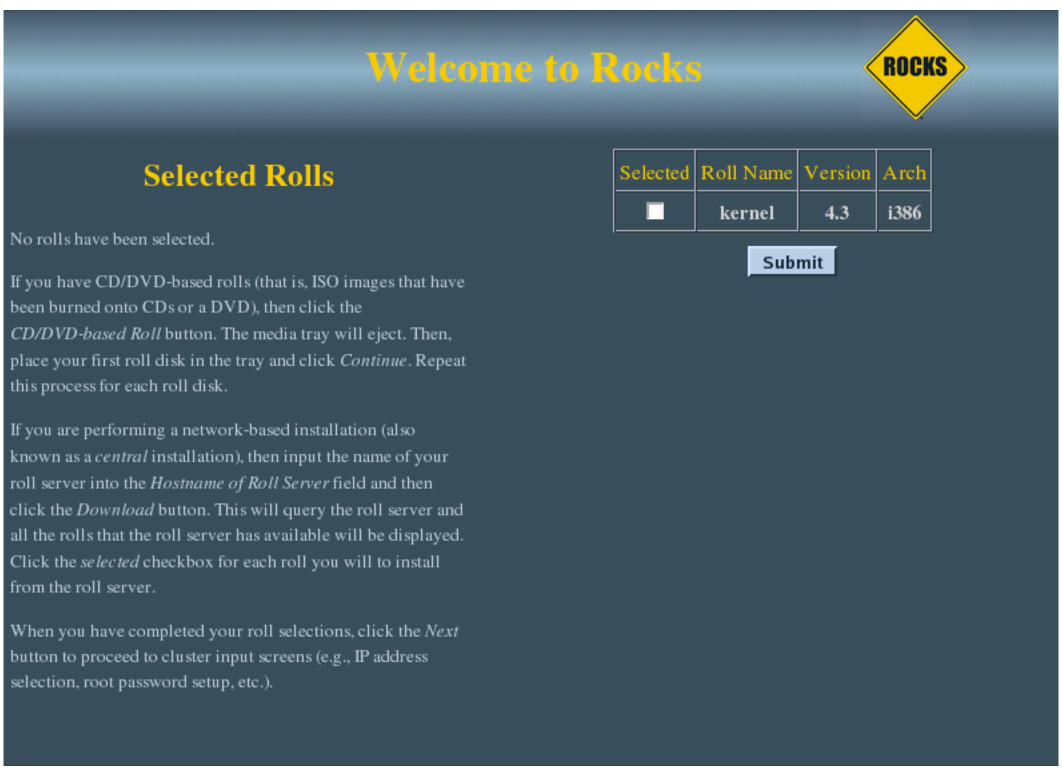
5. The CD will eject and you will see this screen:



Put your first roll in the CD tray (for the first roll, since the Kernel/Boot Roll is already in the tray, simply push the tray back in).

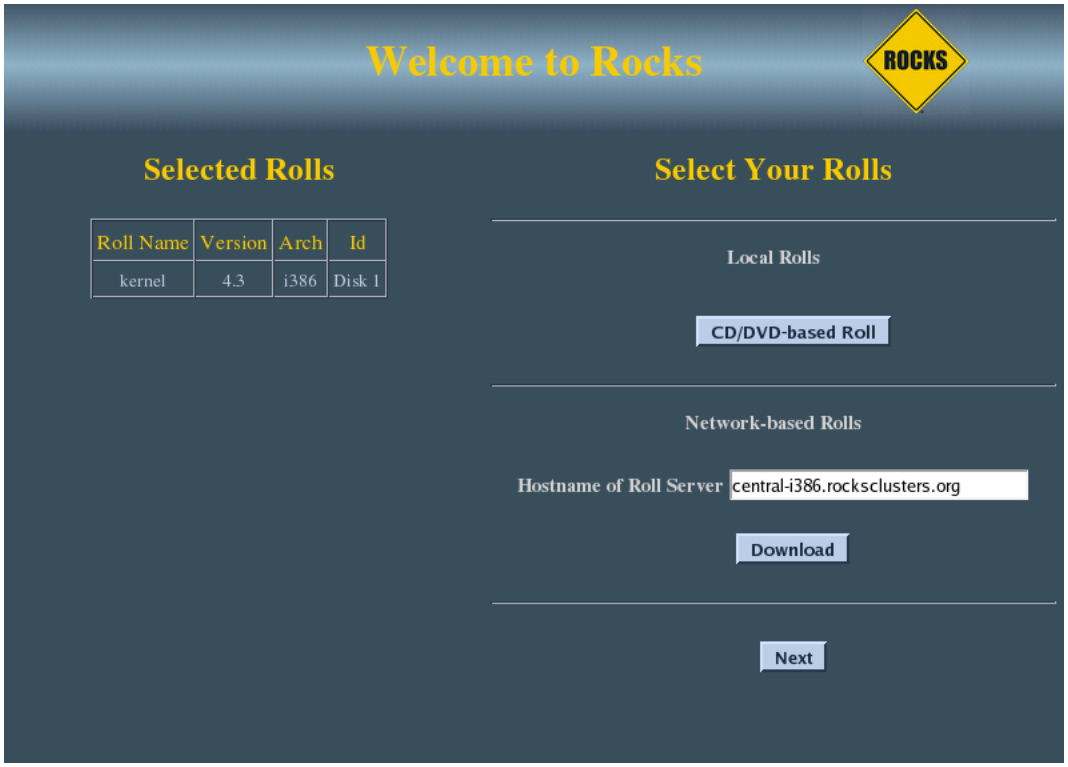
Click the 'Continue' button.

6. The Kernel/Boot Roll will be discovered and display the screen:



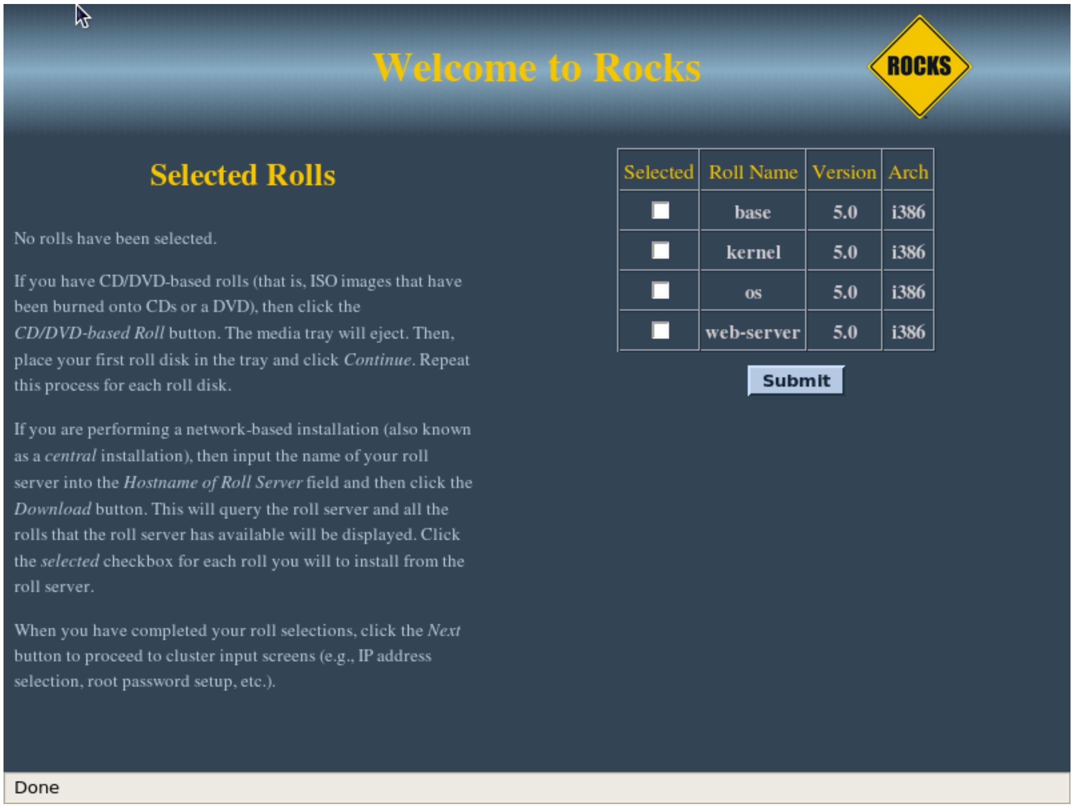
Select the Kernel/Boot Roll by checking the 'Selected' box and clicking the 'Submit' button.

7. This screen shows you have properly selected the Kernel/Boot Roll.



Repeat steps 3-5 for the Base Roll, Web Server Roll and the OS rolls.

8. When you have selected all the rolls associated with a *bare-bones* frontend, the screen should look like:



When you are done with roll selection, click the 'Next' button.

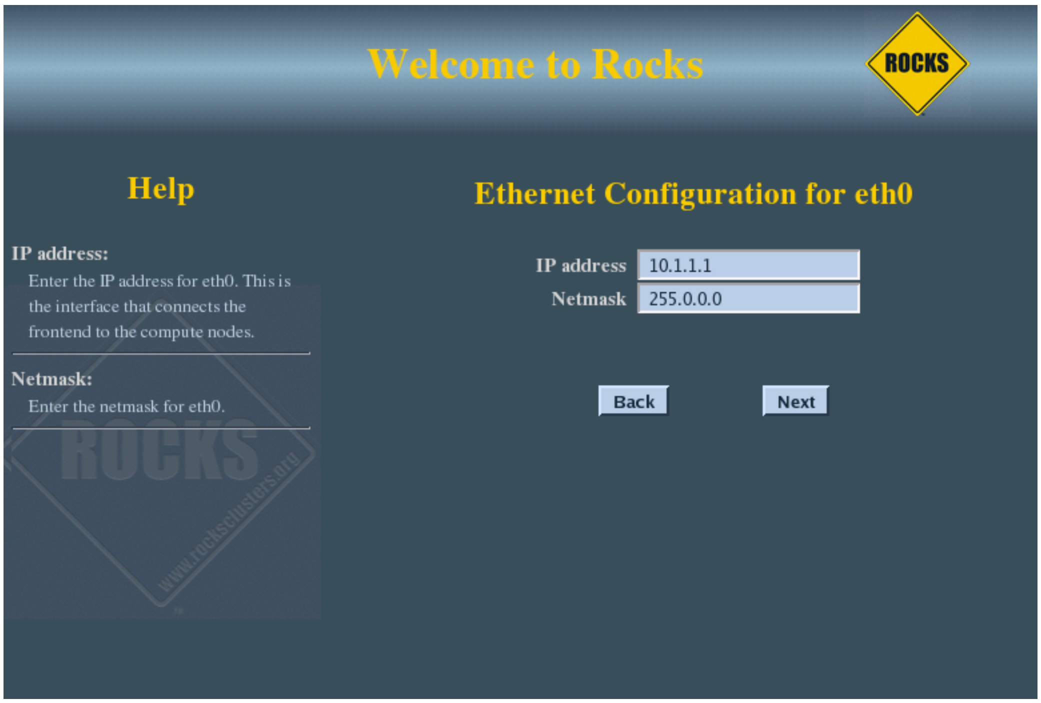
9. Then you'll see the *Cluster Information* screen:

The one important field in this screen is the *Fully-Qualified Host Name* (all other fields are optional).

Choose your hostname carefully. The hostname is written to dozens of files on both the frontend and compute nodes. If the hostname is changed after the frontend is installed, several cluster services will no longer be able to find the frontend machine. Some of these services include: SGE, NFS, AutoFS, and Apache.

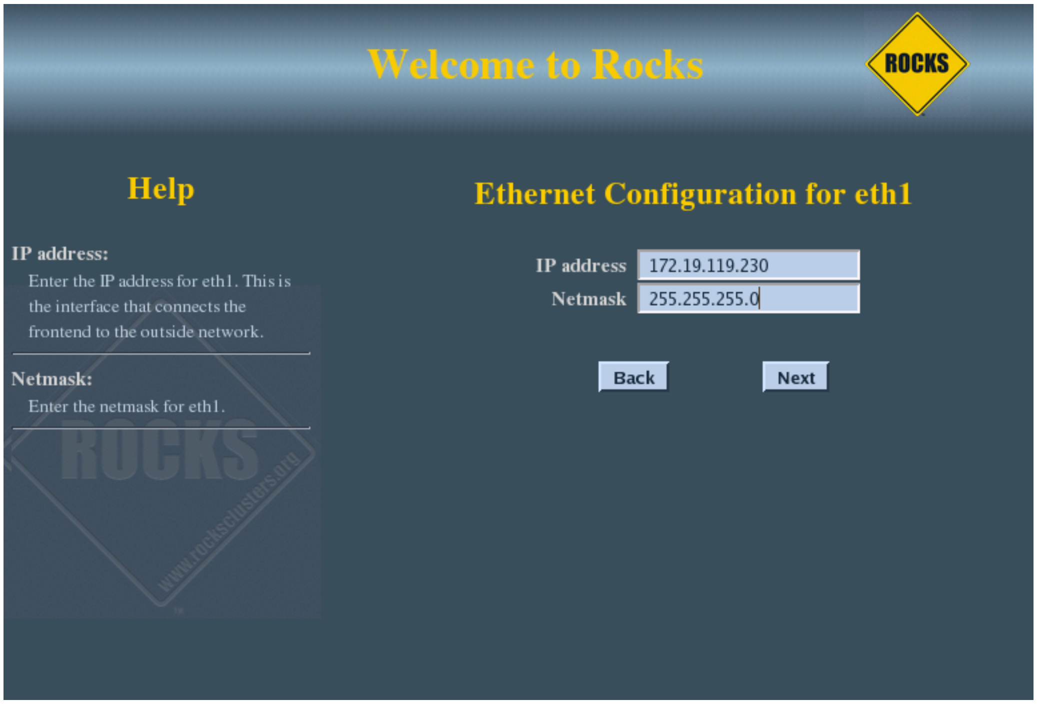
Fill out the form, then click the 'Next' button.

10. The private cluster network configuration screen allows you to set up the networking parameters for the ethernet network that connects the frontend to the compute nodes.



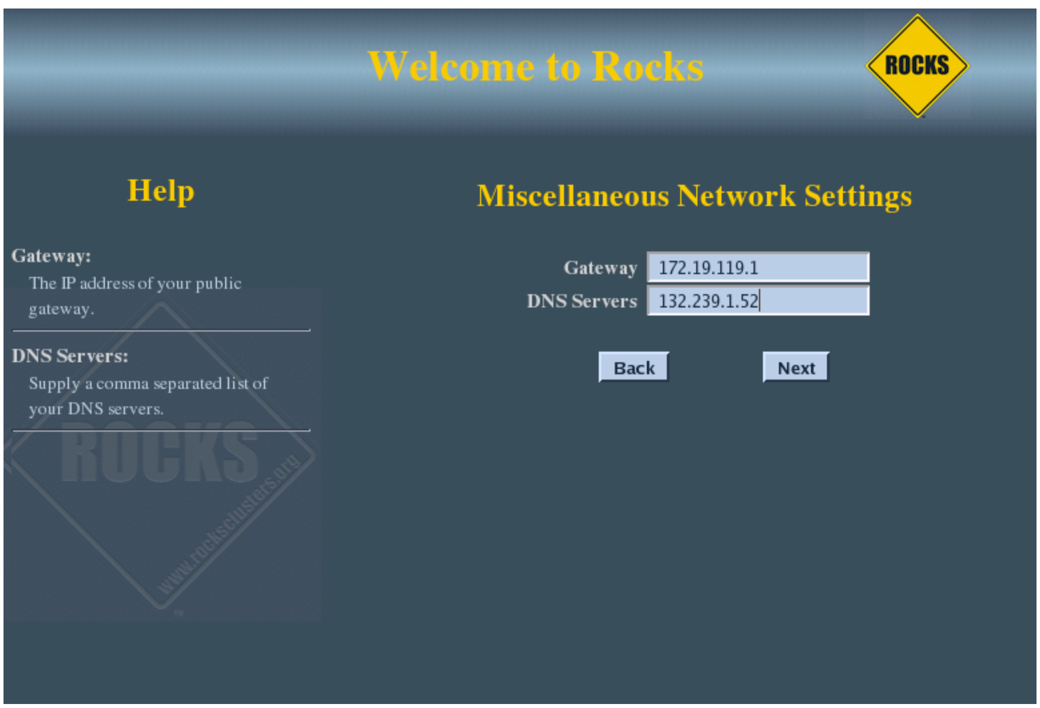
It is recommended that you accept the defaults (by clicking the 'Next' button). But for those who have unique circumstances that requires different values for the internal ethernet connection, we have exposed the network configuration parameters.

11. The public cluster network configuration screen allows you to set up the networking parameters for the ethernet network that connects the frontend to the outside network (e.g., the internet).

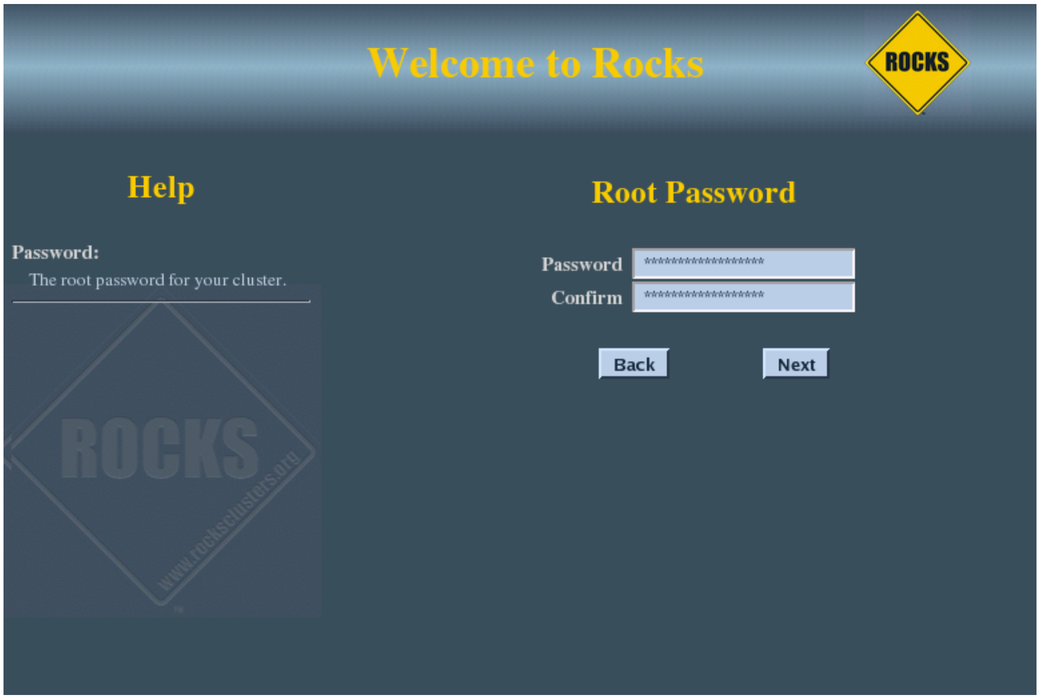


The above window is an example of how we configured the external network on one of our frontend machines.

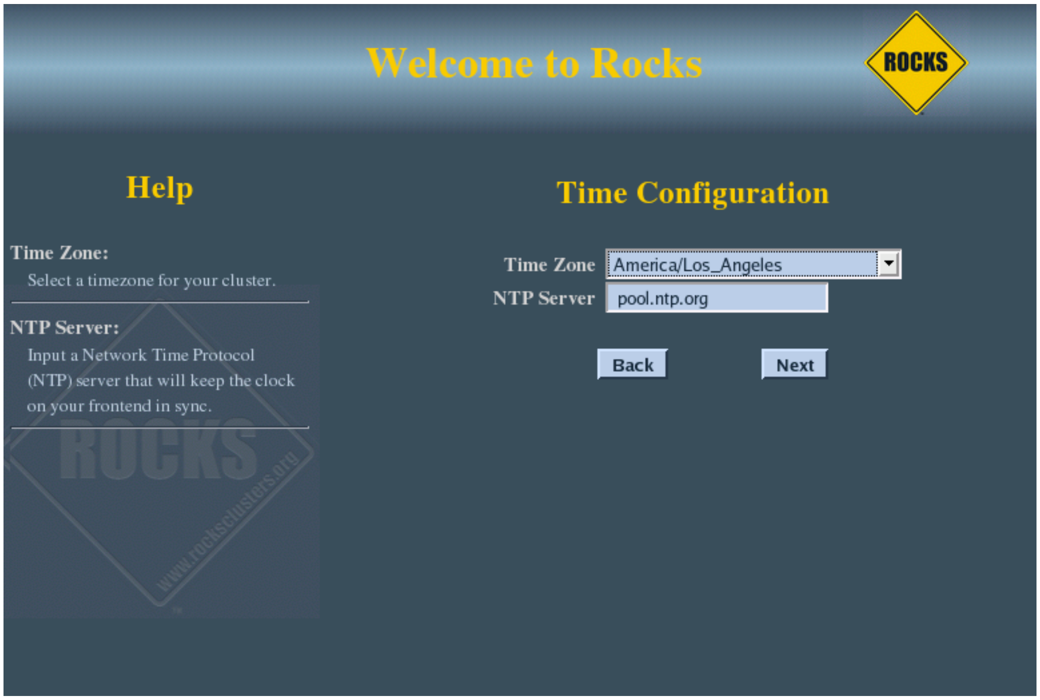
12. Configure the the *Gateway* and *DNS* entries:



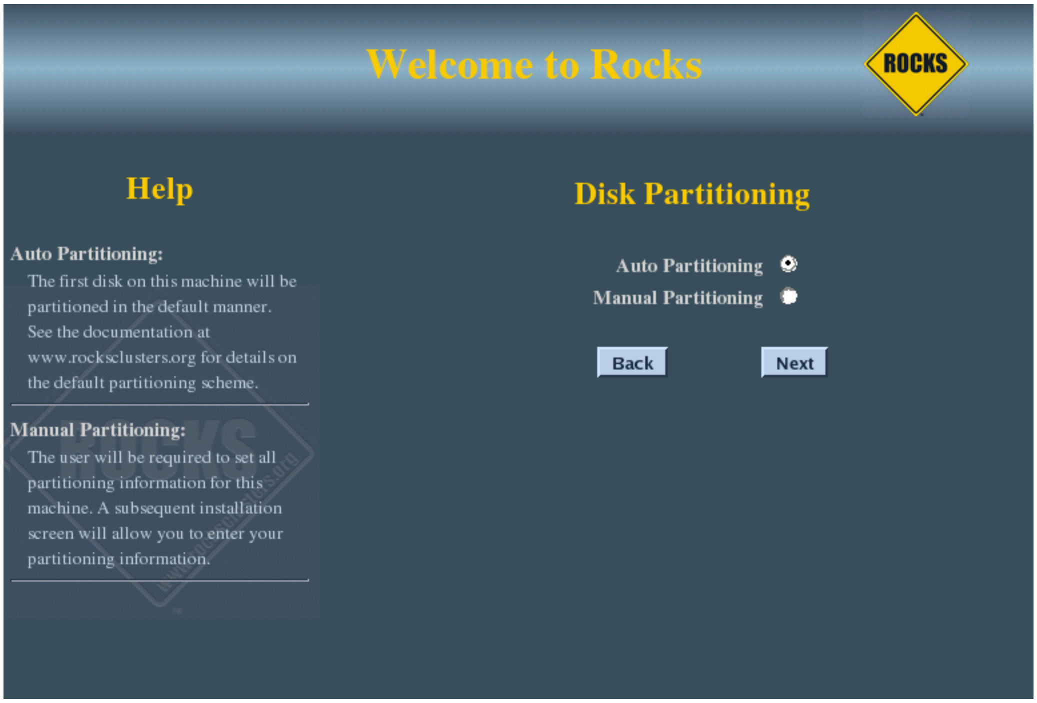
13. Input the root password:



14. Configure the time:



15. The disk partitioning screen allows you to select *automatic* or *manual* partitioning.



To select automatic partitioning, click the `Auto Partitioning` radio button. This will repartition and reformat the first discovered hard drive that is connected to the frontend. All other drives connected to the frontend will be left untouched.

The first discovered drive will be partitioned like:

Table 3-1. Frontend -- Default Root Disk Partition

Partition Name	Size
/	16 GB
/var	4 GB
swap	1 GB
/export (symbolically linked to /state/partition1)	*remainder of root disk*

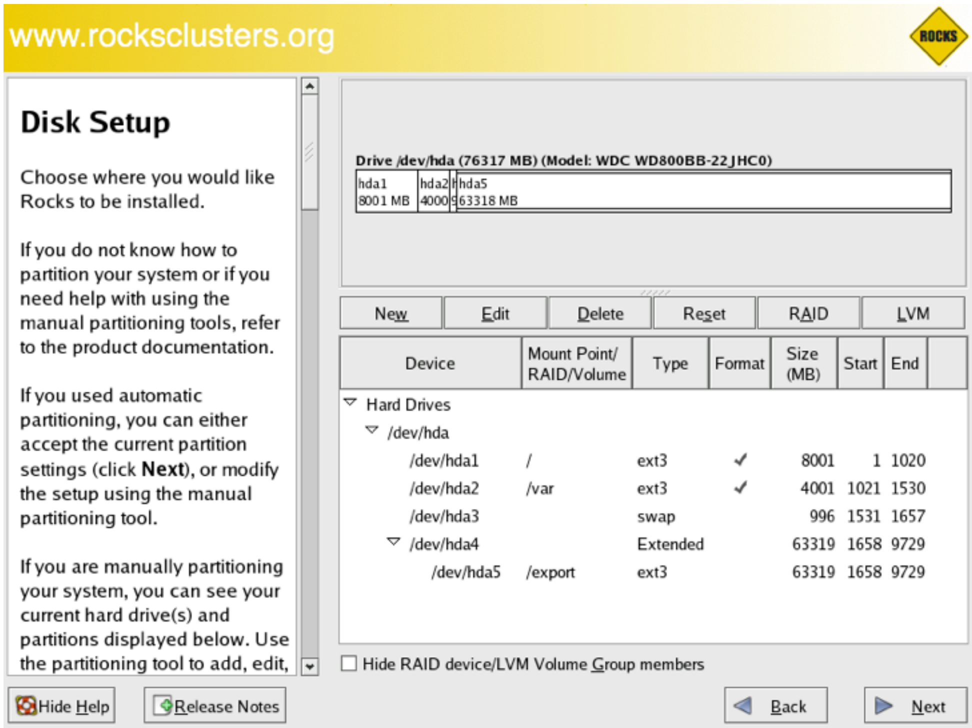
When you use automatic partitioning, the installer will repartition and reformat the *first hard drive* that the installer discovers. All previous data on this drive will be erased. All other drives will be left untouched.

The drive discovery process uses the output of `cat /proc/partitions` to get the list of drives.

For example, if the node has an IDE drive (e.g., "hda") and a SCSI drive (e.g., "sda"), generally the IDE drive is the first drive discovered.

But, there are instances when a drive you don't expect is the first discovered drive (we've seen this with certain fibre channel connected drives). If you are unsure on how the drives will be discovered in a multi-disk frontend, then use manual partitioning.

16. If you selected manual partitioning, then you will now see Red Hat's manual partitioning screen:



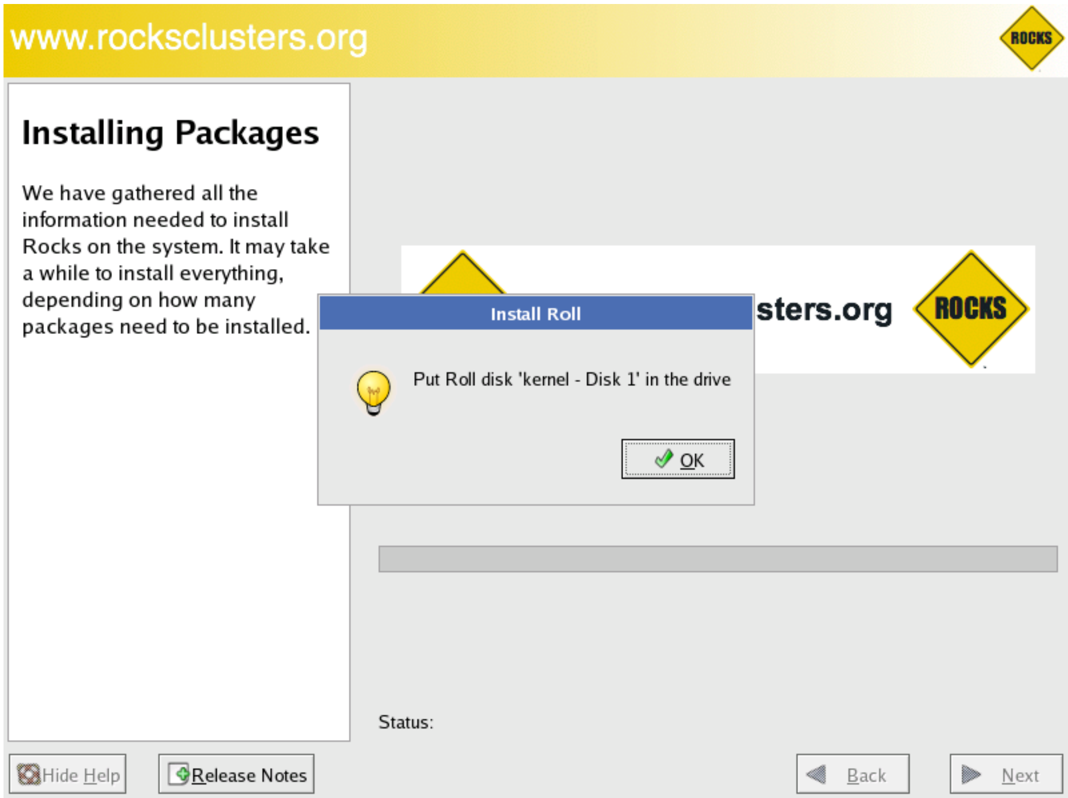
Above is an example of creating a '/', '/var', swap and '/export' partitions.

If you select manual partitioning, you must specify at least 16 GBs for the root partition and you must create a separate `/export` partition.

LVM is not supported by Rocks.

When you finish describing your partitions, click the 'Next' button.

17. The frontend will format its file systems, then it will ask for each of the roll CDs you added at the beginning of the frontend installation.



In the example screen above, insert the Kernel/Boot Roll into the CD tray and click 'OK'.

The contents of the CD will now be copied to the frontend's hard disk.

Repeat this step for each roll you supplied in steps 3-5.

After all the Rolls are copied, no more user interaction is required.

18. After the last roll CD is copied, the packages will be installed:

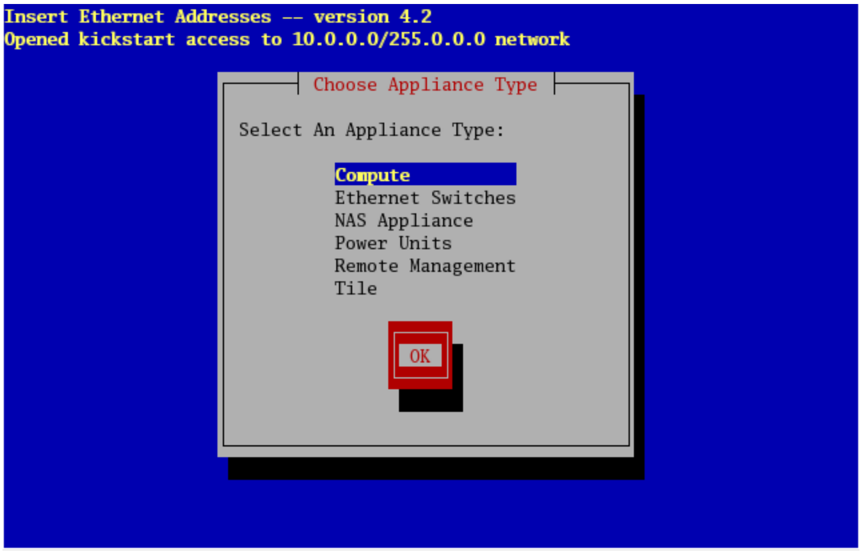
19. Finally, the boot loader will be installed and post configuration scripts will be run in the background. When they complete, the frontend will reboot.

3.3. Install Your Compute Nodes

1. Login to the frontend node as `root`.
2. Run the program which captures compute node DHCP requests and puts their information into the Rocks MySQL database:

   ```
   # insert-ethers
   ```

   This presents a screen that looks like:



If your frontend and compute nodes are connected via a managed ethernet switch, you'll want to select 'Ethernet Switches' from the list above. This is because the default behavior of many managed ethernet switches is to issue DHCP requests in order to receive an IP address that clients can use to configure and monitor the switch.

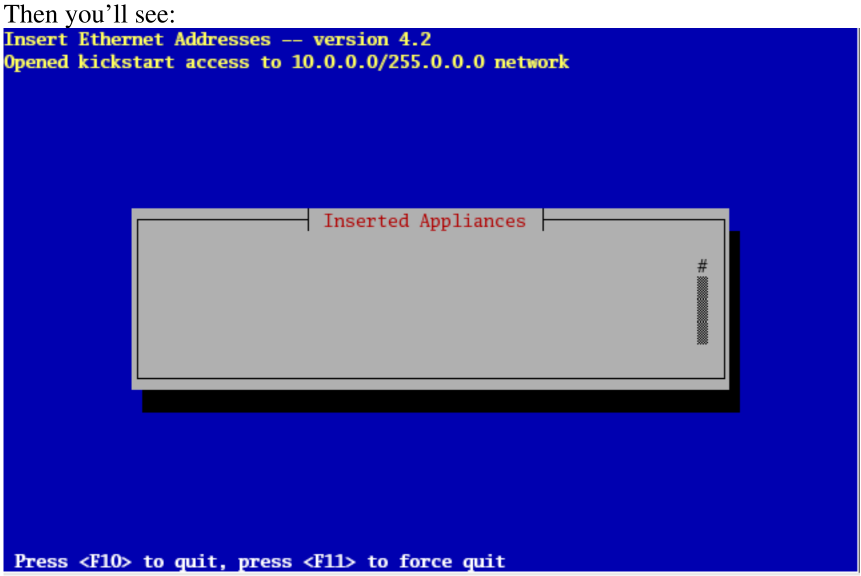
When `insert-ethers` captures the DHCP request for the managed switch, it will configure it as an ethernet switch and store that information in the MySQL database on the frontend.

As a side note, you may have to wait several minutes before the ethernet switch broadcasts its DHCP request. If after 10 minutes (or if `insert-ethers` has correctly detected and configured the ethernet switch), then you should quit `insert-ethers` by hitting the `F8` key.

Now, restart `insert-ethers` and continue reading below to configure your compute nodes.

Take the default selection, `Compute`, hit 'Ok'.

3. Then you'll see:



This indicates that `insert-ethers` is waiting for new compute nodes.

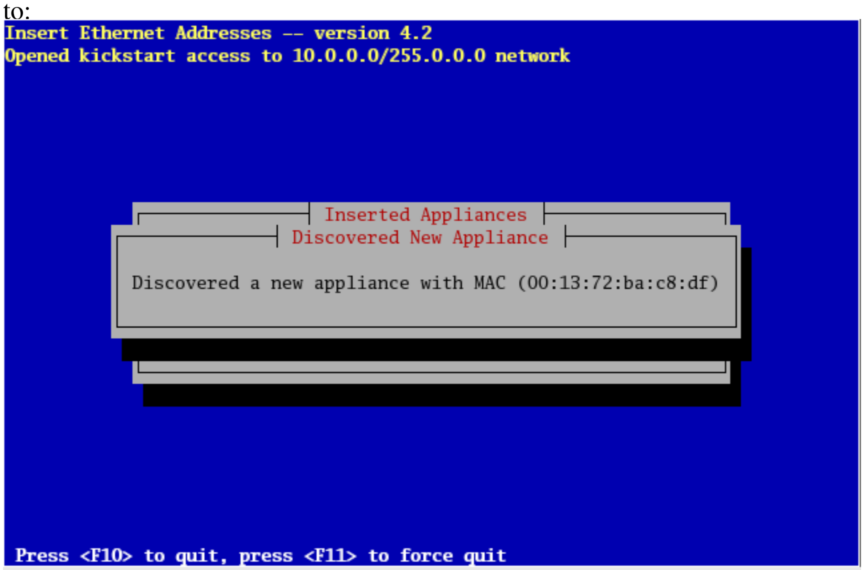
4. Power up the first compute node.

   The BIOS boot order of your compute nodes should be: CD, PXE (Network Boot), Hard Disk.

   If your compute nodes don't support PXE, then you'll need to boot your compute nodes with the Kernel Roll CD.

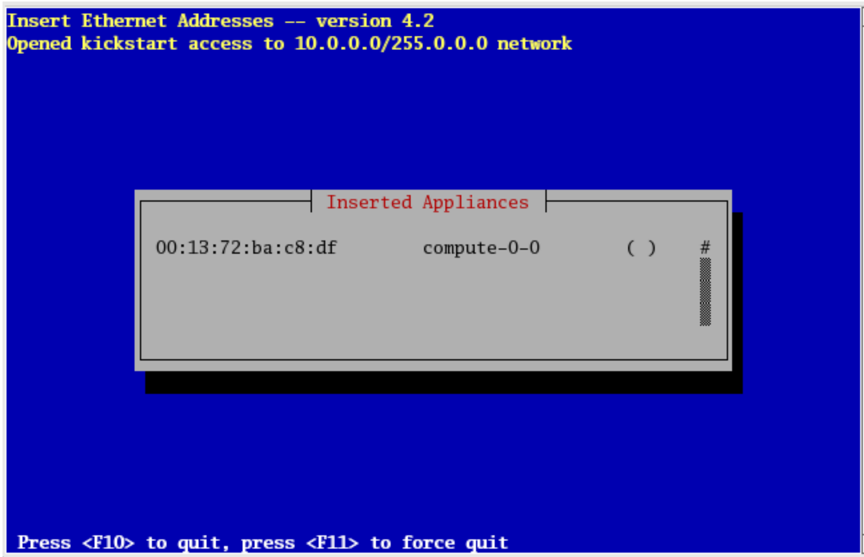
   If you don't have a CD drive in your compute nodes and if the network adapters in your compute nodes don't support PXE, see Using a Floppy to PXE boot.

5. When the frontend machine receives the DHCP request from the compute node, you will see something similar to:



   This indicates that `insert-ethers` received the DHCP request from the compute node, inserted it into the database and updated all configuration files (e.g., `/etc/hosts, /etc/dhcpd.conf` and DNS).

   The above screen will be displayed for a few seconds and then you'll see the following:



In the above image, insert-ethers has discovered a compute node. The "( )" next to compute-0-0 indicates the node has not yet requested a kickstart file. You will see this type of output for each compute node that is successfully identified by insert-ethers.

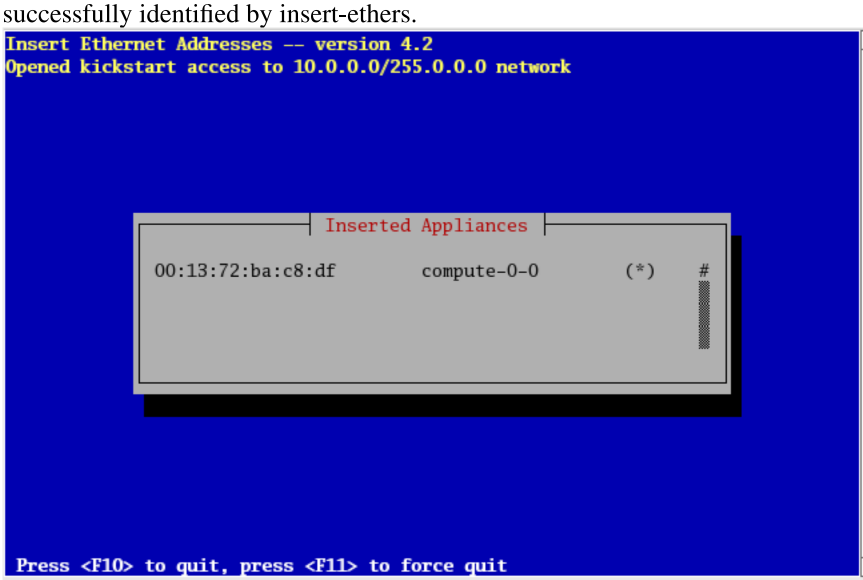


Figure: The compute node has successfully requested a kickstart file from the frontend. If there are no more compute nodes, you may now quit insert-ethers. Kickstart files are retrieved via HTTPS. If there was an error during the transmission, the error code will be visible instead of "*".

6. At this point, you can monitor the installation by using `rocks-console`. Just extract the name of the installing compute node from the `insert-ethers` output (in the example above, the compute node name is `compute-0-0`), and execute:

```
# rocks-console compute-0-0
```

7. After you've installed all the compute nodes in a cabinet, quit `insert-ethers` by hitting the 'F8' key.

8. After you've installed all the compute nodes in the first cabinet and you wish to install the compute nodes in the next cabinet, just start `insert-ethers` like:

```
# insert-ethers --cabinet=1
```

This will name all new compute nodes like `compute-1-0`, `compute-1-1`, ...

3.4. Upgrade or Reconfigure Your Existing Frontend

This procedure describes how to use a Restore Roll to upgrade or reconfigure your existing Rocks cluster.

Let's create a Restore Roll for your frontend. This roll will contain site-specific info that will be used to quickly reconfigure your frontend (see the section below for details).

```
# cd /export/site-roll/rocks/src/roll/restore
# make roll
```

The above command will output a roll ISO image that has the name of the form: *hostname*-restore-*date*-0.*arch*.disk1.iso. For example, on the i386-based frontend with the FQDN of *rocks-45.sdsc.edu*, the roll will be named like:

```
rocks-45.sdsc.edu-restore-2006.07.24-0.i386.disk1.iso
```

Burn your restore roll ISO image to a CD.

Reinstall the frontend by putting the Rocks Boot CD in the CD tray (generally, this is the Kernel/Boot Roll) and reboot the frontend.

At the `boot:` prompt type:

```
build
```

At this point, the installation follows the same steps as a *normal* frontend installation (See the section: Install Frontend) -- with two exceptions:

1. On the first user-input screen (the screen that asks for 'local' and 'network' rolls), be sure to supply the Restore Roll that you just created.
2. You will be forced to manually partition your frontend's root disk.

> You must reformat your `/` partition, your `/var` partition and your `/boot` partition (if it exists).
>
> Also, be sure to assign the mountpoint of `/export` to the partition that contains the users' home areas. Do NOT erase or format this partition, or you will lose the user home directories. Generally, this is the largest partition on the first disk.

After your frontend completes its installation, the last step is to force a re-installation of all of your compute nodes. The following will force a PXE (network install) reboot of all your compute nodes.

```
# ssh-agent $SHELL
# ssh-add
# rocks run host compute '/boot/kickstart/cluster-kickstart-pxe'
```

3.4.1. Restore Roll Internals

By default, the Restore Roll contains two sets of files: system files and user files, and some user scripts. The system files are listed in the 'FILES' directive in the file: `/export/site-roll/rocks/src/roll/restore/src/system-files/version.mk`.

```
FILES           = /etc/passwd /etc/shadow /etc/gshadow /etc/group \
                  /etc/exports /etc/auto.home /etc/motd
```

The user files are listed in the 'FILES' directive in the file: `/export/site-roll/rocks/src/roll/restore/version.mk`.

```
FILES           += /etc/X11/xorg.conf
```

If you have other files you'd like saved and restored, then append them to the 'FILES' directive in the file `/export/site-roll/rocks/src/roll/restore/version.mk`, then rebuild the restore roll.

If you'd like to add your own post sections, you can add the name of the script to the 'SCRIPTS' directive of the `/export/site-roll/rocks/src/roll/restore/version.mk` file.

```
SCRIPTS += /share/apps/myscript.sh /share/apps/myscript2.py
```

This will add the shell script `/share/apps/myscript.sh`, and the python script `/share/apps/myscript2.py` in the post section of the restore-user-files.xml file.

If you'd like to run the script in "nochroot" mode, add

```
# nochroot
```

as the first comment in your script file after the interpreter line, if one is present.

For example

```
#!/bin/bash
#nochroot
echo "This is myscript.sh"
```

or

```
#nochroot
echo "This is myscript.sh"
```

will run the above code in the "nochroot" mode during installation. As opposed to

```
echo "This is myscript.sh"
#nochroot
```

or

```
#!/bin/bash
echo "This is myscript.sh"
```

will NOT run the script under "nochroot" mode.

All the files under `/export/rocks/install/site-profiles` are saved and restored. So, any user modifications that are added via the XML node method will be preserved.

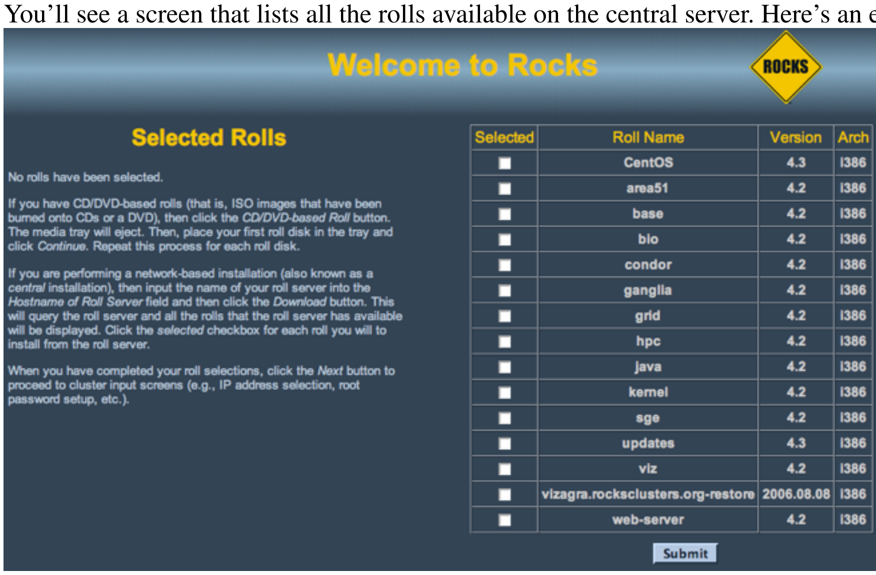
The networking info for all node interfaces (e.g., the frontend, compute nodes, NAS appliances, etc.) are saved and restored. This is accomplished via the ’rocks dump’ command.

3.5. Installing a Frontend over the Network

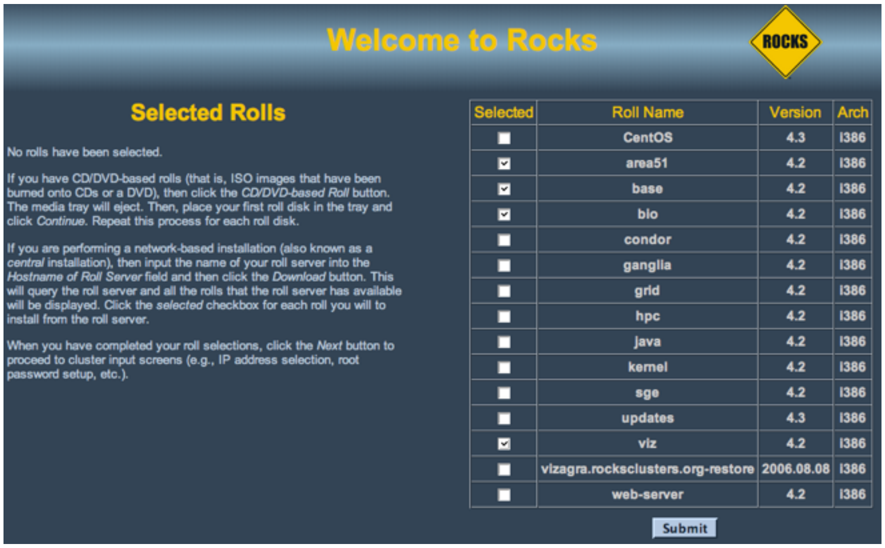
This section describes installing a Rocks frontend from a "Central" server over the wide area network, a process called WAN kickstart. The client frontend will retrieve Rocks Rolls and configuration over the Internet, and use these to install itself.

1. First, boot the node that will be your new frontend with the Kernel/Boot Roll CD (see steps 1 and 2 in the section "Install Frontend").
2. Then you’ll see the screen as described in step 3 in the section "Install Frontend". Enter the FQDN of your central server in the *Hostname of Roll Server* text box (don’t change this value if you want to use the default central server) then and click the `Download` button.

   You’ll see a screen that lists all the rolls available on the central server. Here’s an example:

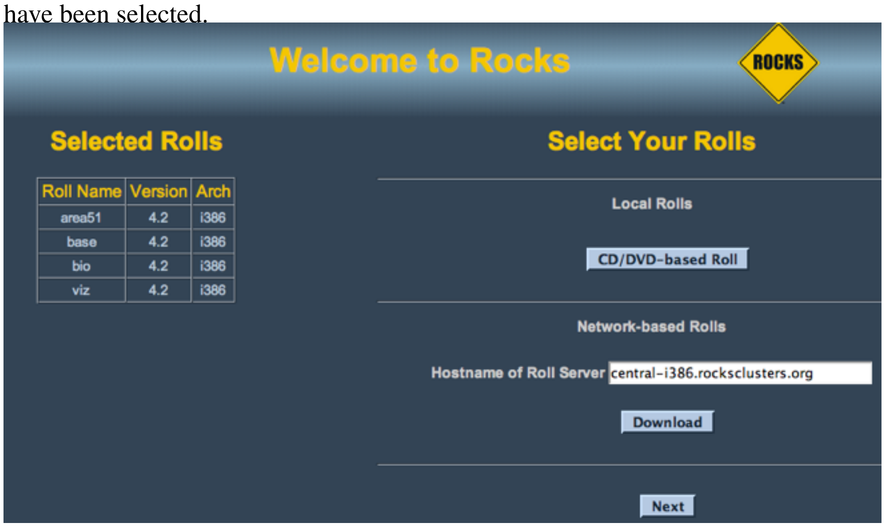


3. Now, select the rolls from the central server. To select a roll, click the checkbox next to roll. For example, this screen shows the *area51*, *base*, *bio* and *viz* rolls selected:



Click the *Submit* button to continue.

4. Now you'll see a screen similar to the screen below. This screen indicates that the *area51*, *base*, *bio* and *viz* rolls have been selected.



5. To select more rolls from another server, go to step 1 and enter a different FQDN.
6. If you'd like to include CD-based rolls with your Network-based rolls, click the *CD/DVD-based Roll* button and follow the instructions in the section "Install Frontend" starting at step 4.
7. When you are finished installing CD-based rolls, you will enter into the familiar Rocks installation windows. These may change depending on what rolls you have selected. Again the section "Install Frontend" has details for this process.
8. The installer will then retrieve the chosen rolls, rebuild the distribution with all rolls included, then install the packages. Finally, the installer will proceed with the post-section and other elements of a standard frontend install.

Your frontend should now be installed and ready to initialize compute nodes (see section Install Compute Nodes).

3.6. Enabling Public Web Access to Your Frontend

To permenantly enable selected web access to the cluster from other machines on the public network, follow the steps below. Apache's access control directives will provide protection for the most sensitive parts of the cluster web site, however some effort will be necessary to make effective use of them.

HTTP (web access protocol) is a clear-text channel into your cluster. Although the Apache webserver is mature and well tested, security holes in the PHP engine have been found and exploited. Opening web access to the outside world by following the instructions below will make your cluster more prone to malicious attacks and breakins.

To open port 80 (the 'www' service) for the public network of frontend, execute:

```
# rocks open host firewall localhost network=public protocol=tcp service=www
```

Then we can see the what the resulting firewall rules will look like:

```
# rocks report host firewall localhost
<file name="/etc/sysconfig/iptables" perms="500">
*nat
-A POSTROUTING -o eth1 -j MASQUERADE
COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A FORWARD -i eth0 -j ACCEPT
-A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i eth0 -j ACCEPT
-A INPUT -i eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -i eth1 -p tcp --dport https -m state --state NEW --source &Kickstart_PublicNetwork;/&Kicks
-A INPUT -i eth1 -p tcp --dport ssh -m state --state NEW -j ACCEPT
-A INPUT -i eth1 -p tcp --dport www -m state --state NEW --source &Kickstart_PublicNetwork;/&Kicksta
-A INPUT -i eth1 -p tcp --dport www -j ACCEPT
# block mysql traffic from non-private interfaces
-A INPUT -p tcp --dport 3306 -j REJECT
# block foundation mysql traffic from non-private interfaces
-A INPUT -p tcp --dport 40000 -j REJECT
# block ganglia traffic from non-private interfaces
-A INPUT -p udp --dport 8649 -j REJECT
-A INPUT -p tcp --dport 0:1024 -j REJECT
-A INPUT -p udp --dport 0:1024 -j REJECT
COMMIT
</file>
```

In the above example, eth0 is associated with the private network and eth1 is associated with the public network.

Notice the line: "-A INPUT -i eth1 -p tcp --dport www -j ACCEPT". This is the line in the firewall configuration that will allow web traffic from any source to flow in and out of the frontend. This line was added to your firewall configuration with the "rocks open host firewall" command that you executed.

Also, notice the line: "-A INPUT -i eth1 -p tcp --dport www -m state --state NEW --source &Kickstart_PublicNetwork;/&Kickstart_PublicNetmask; -j ACCEPT". This default Rocks firewall rule allows web traffic from *your local public subnet* to flow in and out of the frontend.

Now apply the configuration to the host:

```
# rocks sync host firewall localhost
```

The host will now accept web traffic on its public interface.

Test your changes by pointing a web browser to `http://my.cluster.org/`, where "my.cluster.org" is the DNS name of your frontend machine.

If you cannot connect to this address, the problem is most likely in your network connectivity between your web browser and the cluster. Check that you can ping the frontend machine from the machine running the web browser, that you can ssh into it, etc.

Notes

1. http://www.oreilly.com

Chapter 4. Defining and Modifying Networks and Network Interfaces

4.1. Networks, Subnets, VLANs and Interfaces

Rocks uses a SQL database to hold information about nodes including network device information. In version 5.1 support was added for VLAN tagged interfaces to enable construction of virtual clusters and other more complicated network scenarios. There are a large number of commands that allow manipulation of subnet definitions, physical interfaces, and logical VLAN interfaces.

> The basic model of operation is for an administrator to use a series of commands to add and set/change networking definitions in the database and then either re-install a node or reconfigure/restart the network configuration by calling rocks sync config <host>.

4.2. Named Networks/Subnets

Rocks clusters are required to have two subnets defined: "public" and "private", but a cluster owner can define more subnets. The commmand rocks list network lists the defined networks

```
[root@rocks ~]# rocks list network
NETWORK    SUBNET       NETMASK
private:   172.16.254.0 255.255.255.0
public:    132.239.8.0  255.255.255.0
optiputer: 67.58.32.0   255.255.224.0
```

In the screen above, the additional network called "optiputer" is defined with netmask 255.255.224.0(/19). To add a network called "fast" as 192.168.1.0 and netmask 255.255.255.0(/24) do the following

```
[root@rocks ~]# rocks add network fast subnet=192.168.1.0 netmask=255.255.255.0
[root@rocks ~]# rocks list network
NETWORK    SUBNET       NETMASK
private:   172.16.254.0 255.255.255.0
public:    132.239.8.0  255.255.255.0
optiputer: 67.58.32.0   255.255.224.0
fast:      192.168.1.0  255.255.255.0
```

The subnet and netmask of an existing network can be changed using rocks set network subnet and rocks set network netmask commands.

4.3. Host Interfaces

There are three types of interfaces that a cluster owner may need to be concerned about: physical, logical, and VLAN (virtual LAN) bridges. Linux (and other OSes like Solaris) support logical interfaces that share a particular physical network port. The following shows physical network devices and associations of those devices to a named network (or subnet, used interchangably in this discussion). In the figures below, the /<nn> notation is a standard method of how to specify the number of bits in the netmask. Examples include: /24=255.255.255.0 (Class C subnet), /16=255.255.0.0 (Class B subnet), /8=255.0.0.0 (Class A subnet) and /25=255.255.255.128

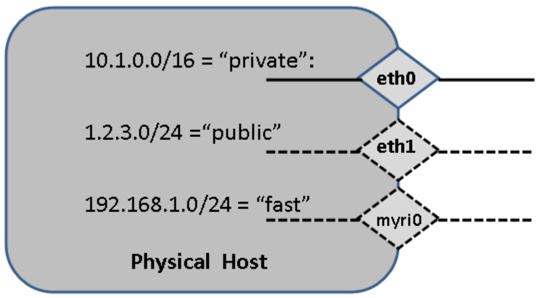


FIGURE: Hosts can have any number of physical networking devices. Every Rocks node must have a private network defined (e.g., eth0). Frontends also must have a separate public network (e.g., eth1). Other devices could be myri0 (for Myrinet) or ib0 (for Infiniband).

Adding a new network interface to a host can be done from the command line. For example, to add an interface named "myri0" with IP address 192.168.1.10 on the logical subnet "fast":

```
[root@rocks ~]# rocks add host interface compute-0-0-1 iface=myri0 subnet=fast ip=192.168.1.10
[root@rocks ~]# rocks list host interface compute-0-0-1
SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME          VLAN
private eth0  00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0-0-1 ------
fast    myri0 ----------------- 192.168.1.10   255.255.255.0 ------ ------------- ------
```

You can also set other fields for a host interface (if the field is one of [mac, ip, module, name, vlan]) with the command `rocks set host interface <field> <host> iface=<iface> value`. To set the name associated with the myri0 interface to compute-myri-0-0-1 on the node compute-0-0-1, execute:

```
[root@rocks ~]# rocks set host interface name compute-0-0-1 iface=myri0 compute-myri-0-0-1
[root@rocks ~]# rocks list host interface compute-0-0-1
SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME               VLAN
private eth0  00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0-0-1      ------
fast    myri0 ----------------- 192.168.1.10   255.255.255.0 ------ compute-myri-0-0-1 ------
```

4.4. Virtual LANs (VLANs) and Logical VLAN Bridges

Linux supports VLAN tagging on virtual interfaces (i.e., IEEE 802.1Q). For example, if a host has physical interface eth0 (untagged), then the kernel can send and receive a tagged packets if a properly defined interface named eth0.<vlan> has been created and properly configured. Tagging allows the same physical network to be partitioned into many different networks. A key feature of VLAN tagging is that a broadcast packet (e.g. a DHCPDISCOVER packet) only broadcasts on the tagged VLAN in which is was initially sent.

Rocks supports two types of VLAN interfaces - the first is an explicit device name like eth0.10 that is defined on a particular physical interface. The second is a logical device name of the form "vlan*". In Rocks, the physical VLAN device can also have an IP address associated with it, however a logical VLAN device cannot. We use logical VLANs to construct bridges suitable for virtual clusters.

1. Explicit VLAN Devices of the form <interface>.<vlan> can have IP addresses assigned
2. Rocks-Specific: Logical VLAN Devices of the form "vlan*" CANNOT have IP address assigned

4.4.1. Physical VLAN Devices

Physical VLAN devices are interfaces associated with specific physical interfaces. While eth0 is used as an example, any physical IP interface can have a VLAN associated with it.

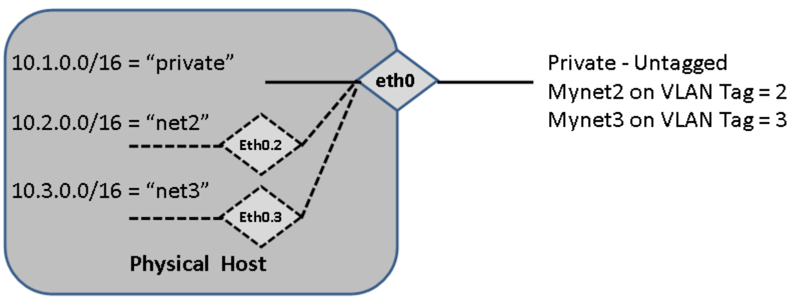


FIGURE: Physical VLAN device called eth0.2. This device may be assigned an IP and a network name (e.g. "net") that is unrelated to the network name of the physical device (eth0). All packets sent on this interface will be tagged with VLAN=2. Multiple Physical VLAN devices can be defined.

Use the following example to add a physical VLAN device, assign a tag, and an IP address:

```
[root@rocks ~]# rocks add host interface compute-0-0-1 iface=eth0.2 subnet=net2 ip=10.2.1.10
[root@rocks ~]# rocks set host interface vlan compute-0-0-1 iface=eth0.2 vlan=2
[root@rocks ~]# rocks list host interface compute-0-0-1
SUBNET  IFACE  MAC               IP             NETMASK       MODULE NAME          VLAN
private eth0   00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0-0-1 ------
net2    eth0.2 ----------------- 10.2.1.10      255.255.255.0 ------ ------------- 2
```

4.4.2. Logical VLAN Devices

The second kind of VLAN interface that Rocks supports is what we call a *logical VLAN device*. The Virtual VLAN gives the ability of having a raw interface with no IP address assigned that is generally used as a bridge for virtual machines. Virtual VLAN devices have their subnet=<subnet of physical>

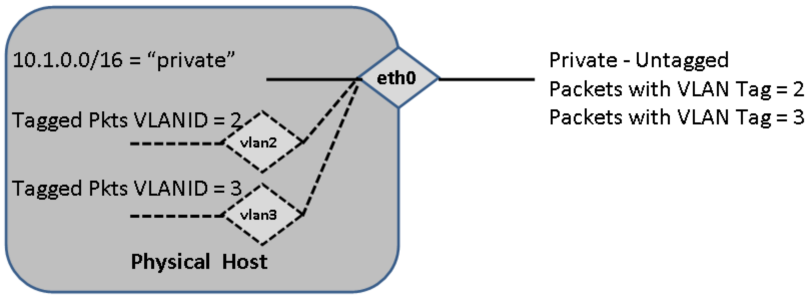


FIGURE: Virtual VLAN devices called vlan2 and vlan3. These types of devices may NOT have an IP address (This is a Rocks-specific construction).

```
[root@rocks ~]# rocks add host interface compute-0-0-1 vlan2
[root@rocks ~]# rocks add host interface compute-0-0-1 vlan3
[root@rocks ~]# rocks set host interface vlan compute-0-0-1 vlan2 2
[root@rocks ~]# rocks set host interface vlan compute-0-0-1 vlan3 3
[root@rocks ~]# rocks list host interface compute-0-0-1
SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME          VLAN
private eth0  00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0-0-1 ------
------- vlan2 ----------------- -------------- ------------- ------ ------------- 2
------- vlan3 ----------------- -------------- ------------- ------ ------------- 3
```

At this stage, the vlan interfaces are not currently associated with any physical network device. Linux will not configure these devices on the node without the association. We overload the meaning of subnet in this case to mean: "associate the logical vlan device with the physical device that is in subnet 'x'". As an example, we can associate both vlan2 and vlan3 to be tagged packet interfaces on the the subnet named private.

```
[root@tranquil ~]# rocks set host interface subnet compute-0-0-1 vlan2 subnet=private
[root@tranquil ~]# rocks set host interface subnet compute-0-0-1 vlan3 subnet=private
[root@tranquil ~]# rocks list host interface compute-0-0-1
SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME          VLAN
private eth0  00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0-0-1 ------
private vlan2 ----------------- -------------- ------------- ------ ------------- 2
private vlan3 ----------------- -------------- ------------- ------ ------------- 3
```

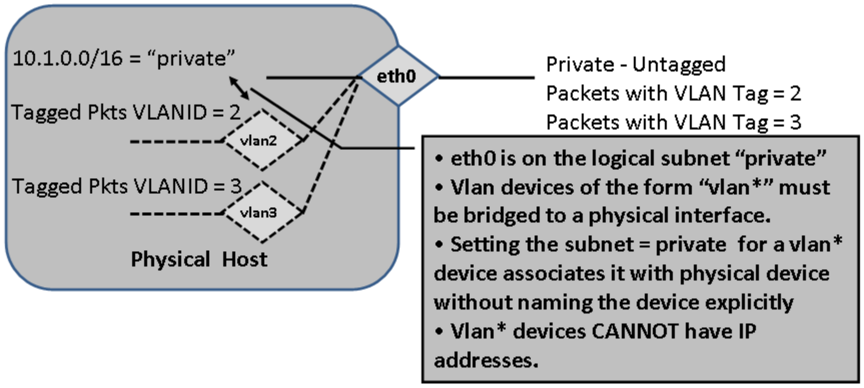


FIGURE: Virtual VLAN devices called vlan2 and vlan3 are associated with the physical device that is desigated as subnet private. Notice, that no netmask is associated with the vlan2 and vlan3 devices. These are raw, tagged packet interfaces and are mostly used for bridges when hosting VMs.

4.5. Network Bridging for Virtual Machines

Rocks support of Virtual Machines requires the proper setup of networking bridges. Rocks supports multiple network adapters for Virtual Machines, In this section, we describe the various kinds of bridging scenarios for virtual machines and how to set them up. For these examples, the physical machine will be called vm-container-0-0,

4.5.1. VM Network Bridging to Physical Devices

When a VM is bridged to the physical device, it must be assigned in the same subnet as the physical device with a compatible IP address

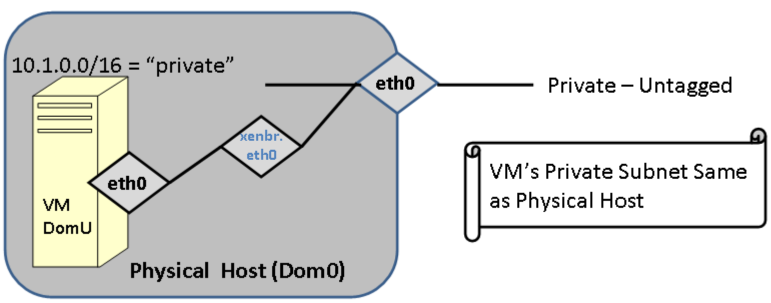


FIGURE: The Virtual machine is bridged to eth0. In this case eth0 of the VM is in the same subnet (with a compatible IP) address. The VM and the container will be able to ping each other. This was the only configuration supported in Rocks 5.0

The following example shows this most basic of bridging scenarios. The guest (compute-0-0-1) and the container (vm-container-0-0) are in the same IP subnet and will be able to ping each other.

```
[root@tranquil images]# rocks list host interface vm-container-0-0 compute-0-0-1
HOST                SUBNET    IFACE    MAC                  IP              NETMASK       MODULE NAME
```

```
compute-0-0-1:     private   eth0     00:16:3e:00:00:11 172.16.254.192 255.255.255.0 xennet compute-0
vm-container-0-0: private   eth0     00:09:6b:89:39:68 172.16.254.238 255.255.255.0 e1000  vm-contai
```

4.5.2. Logical VLAN Devices

In this scenario, The guest (hosted-vm-0-0-2) and the host (vm-container-0-0) are not in the same logical network.

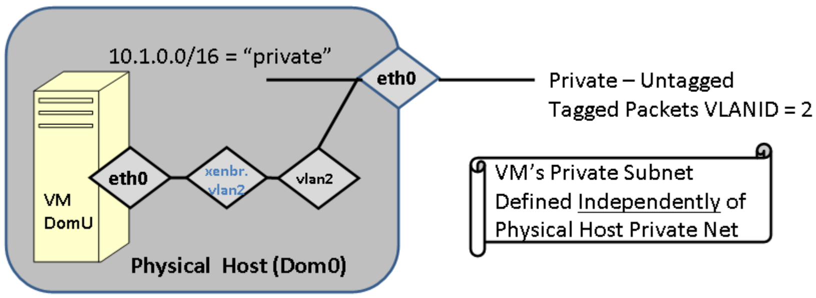


FIGURE: Guest VM is bridged through a logical VLAN device.

```
[root@rocks ~]# rocks list host interface vm-container-0-0 hosted-vm-0-0-0
HOST              SUBNET  IFACE MAC               IP             NETMASK   MODULE NAME             V
hosted-vm-0-0-0:  ------- eth0  00:16:3e:00:00:05 -------------- --------- ------ hosted-vm-0-0   2
vm-container-0-0: private eth0  00:0e:0c:5d:7e:5e 10.255.255.254 255.0.0.0 e1000  vm-container-0-0 -
vm-container-0-0: private vlan2 ----------------- -------------- --------- ------ ---------------- 2
```

In the above configuration, Logical VLAN device vlan2 (with tag=2) will be on the physical network eth0 on vm-container-0-0. The hosted-vm-0-0-0 (a Rocks "appliance" that simply holds a generic VM guest) will have have its interface on VLAN=2. The physical machine must have a Logical vlan device with the same tag.

Below we give a more complicated configuration and walk through exactly what is bridged where.

```
[root@rocks ~]# rocks list host interface vm-container-0-0
SUBNET  IFACE   MAC               IP             NETMASK       MODULE NAME             VLAN
private eth0    00:0e:0c:5d:7e:5e 10.255.255.254 255.0.0.0     e1000  vm-container-0-0 ------
net10   eth1    00:10:18:31:74:84 192.168.1.10   255.255.255.0 tg3    vm-net10-0-0     ------
net10   vlan100 ----------------- -------------- ------------- ------ ---------------- 100
private vlan2   ----------------- -------------- ------------- ------ ---------------- 2

[root@rocks ~]# rocks list host interface hosted-vm-0-0-0
SUBNET IFACE MAC               IP NETMASK MODULE NAME            VLAN
------ eth0  00:16:3e:00:00:05 -- ------- ------ hosted-vm-0-0-0 2
------ eth1  00:16:3e:00:00:80 -- ------- ------ --------------- 100
```

In the above scenario, if hosted-vm-0-0-0 (Xen guest, DomU) were to be booted on physical host vm-container-0-0 (Dom0), the packets from the guest on eth0 will be tagged with VLAN=2, and eth1 with VLAN=100. The host machine must have Logical VLAN interfaces called "vlan*.". To make the proper bridge configuration, Rocks will match the VLANs of the guest interfaces to the VLANs on the host. On the host, logical interface vlan2 is labeled as being on the private network (eth0) and logical vlan100 is labeled as being on the net10 network (eth1).

4.5.3. Networking for Virtual Clusters

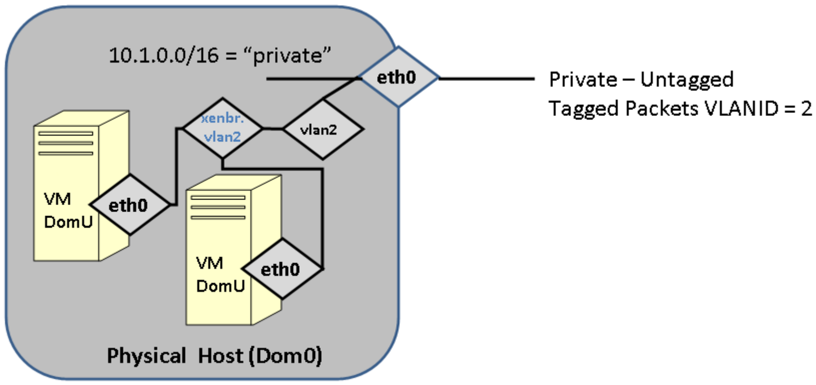


FIGURE: Multiple VMs communicating on a Logical VLAN interface.

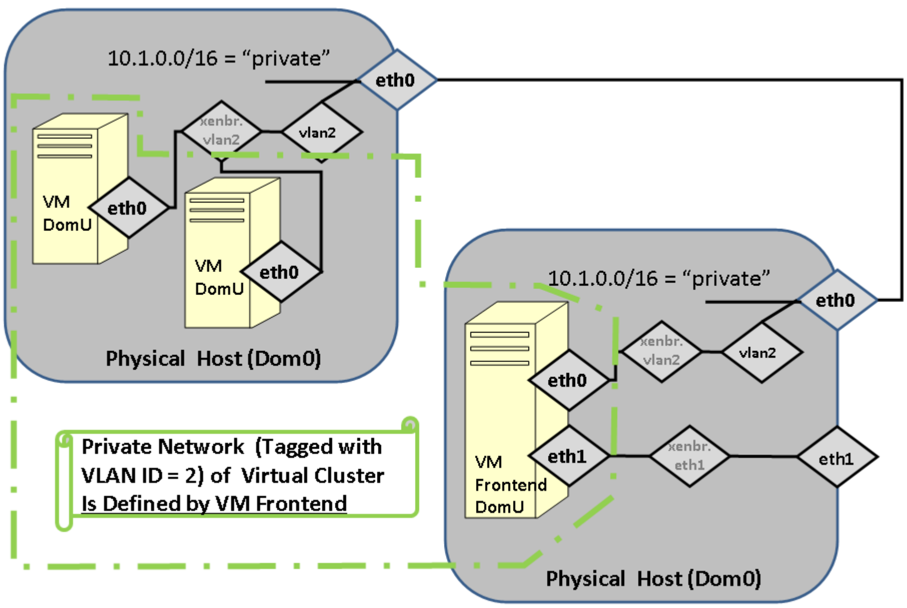


FIGURE: Fully Virtualized cluster, including a virtual frontend.

4.6. Networking Configuration Examples

In this section, we describe some common networking configurations and how to use Rocks commands to set up various networking scenarios.

4.6.1. Adding a public IP address to the second ethernet adapter on a compute node

Often, owners want the second ethernet adapter to be on the public network and for the default routing to be in the public network. Assuming that the public network is 1.2.3.0/255.255.255.0 and the default gateway for that network is 1.2.3.1, the following set of commands define the second interface of a compute to have address 1.2.3.25 with name mypublic.myuniversity.edu, update all required configuration files on the frontend, update all required configuration files on the node compute-0-0 and restart the network on compute-0-0.

```
# rocks set host interface ip compute-0-0 iface=eth1 ip=1.2.3.25
# rocks set host interface name compute-0-0 iface=eth1 name=mypublic.myuniversity.edu
# rocks set host interface subnet compute-0-0 eth1 public
# rocks add host route compute-0-0 1.2.3.0 eth1 netmask=255.255.255.0
# rocks sync config
# rocks sync host network compute-0-0
```

4.6.2. Adding an IP network for local message passing.

Often, users will want to use the second ethernet device for messaging passing. In this example, we illustrate creating a named subnet and then scripting the IP assignment for a rack of 32 nodes with IP range of 192.168.1.10 ... 192.168.1.41.

```
rocks add network fast subnet=192.168.1.0 netmask=255.255.255.0
IP=10
NNODES=32
NODE=0
while [ $NODE -lt $NNODES ]; do \
 rocks set host interface ip compute-0-$NODE iface=eth1 ip=192.168.1.$IP; \
 rocks set host interface subnet compute-0-$NODE iface=eth1 subnet=fast; \
 rocks set host interface name compute-0-$NODE iface=eth1 name=compute-fast-0-$NODE; \
 rocks set host interface subnet compute-0-$NODE eth1 public
 let IP++; \
 let NODE++; \
done
rocks sync config
rocks sync host network compute
```

The above will add the named subnet called "fast", assign IP addresses sequentially, name the eth1 interface on each node, rewrite the DNS configuration (sync config) and finally rewrite and restart the network configuration on each compute appliance. This additional network configuration is persistent across re-installation of nodes.

Chapter 5. Customizing your Rocks Installation

5.1. Adding Packages to Compute Nodes

Put the package you want to add in:

```
/export/rocks/install/contrib/5.4.3/arch/RPMS
```

Where *arch* is your architecture ("i386" or "x86_64").

Create a new XML configuration file that will *extend* the current `compute.xml` configuration file:

```
# cd /export/rocks/install/site-profiles/5.4.3/nodes
# cp skeleton.xml extend-compute.xml
```

Inside `extend-compute.xml`, add the package name by changing the section from:

```
<package> <!-- insert your package name here --> </package>
```

to:

```
<package> your package  </package>
```

It is important that you enter the *base name* of the package in `extend-compute.xml` and not the full name.

For example, if the package you are adding is named *XFree86-100dpi-fonts-4.2.0-6.47.i386.rpm*, input *XFree86-100dpi-fonts* as the package name in `extend-compute.xml`.

```
<package>XFree86-100dpi-fonts</package>
```

If you have multiple packages you'd like to add, you'll need a separate `<package>` tag for each. For example, to add both the 100 and 75 dpi fonts, the following lines should be in `extend-compute.xml`:

```
<package>XFree86-100dpi-fonts</package>
<package>XFree86-75dpi-fonts</package>
```

Also, make sure that you remove any package lines which do not have a package in them. For example, the file should NOT contain any lines such as:

```
<package> <!-- insert your package name here --> </package>
```

Now build a new Rocks distribution. This will bind the new package into a RedHat compatible distribution in the directory */export/rocks/install/rocks-dist/...*.

```
# cd /export/rocks/install
# rocks create distro
```

Now, reinstall your compute nodes.

5.1.1. Adding Specific Architecture Packages to Compute Nodes

Often on x86_64-based clusters, one wants to add the x86_64 and i386 version of a package to compute nodes. To do this, in your `extend-compute.xml` file, supply the section:

```
<package>pkg.x86_64</package>
<package>pkg.i386</package>
```

Where *pkg* is the basename of the package.

Now build a new Rocks distribution.

```
# cd /export/rocks/install
# rocks create distro
```

Now, reinstall your compute nodes.

5.2. Customizing Configuration of Compute Nodes

Create a new XML configuration file that will *extend* the current `compute.xml` configuration file:

```
# cd /export/rocks/install/site-profiles/5.4.3/nodes/
# cp skeleton.xml extend-compute.xml
```

Inside `extend-compute.xml`, add your configuration scripts that will be run in the *post configuration* step of the Red Hat installer.

Put your bash scripts in between the tags `<post>` and `</post>`:

```
<post>
 < !-- insert your scripts here -->
</post>
```

To apply your customized configuration scripts to compute nodes, rebuild the distribution:

```
# cd /export/rocks/install
# rocks create distro
```

Then, reinstall your compute nodes.

5.3. Adding Applications to Compute Nodes

If you have code you'd like to share among the compute nodes, but your code isn't in an RPM (or in a roll), then this procedure describes how you can share it with NFS.

On the frontend, go to the directory `/share/apps`.

```
# cd /share/apps
```

Then add the files you'd like to share within this directory.

All files will also be available on the compute nodes under: `/share/apps`. For example:

```
# cd /share/apps
# touch myapp
# ssh compute-0-0
# cd /share/apps
# ls
myapp
```

5.4. Configuring Additional Ethernet Interfaces

For compute nodes, Rocks uses the first ethernet interface (`eth0`) for management (e.g., reinstallation), monitoring (e.g., Ganglia) and message passing (e.g., OpenMPI over ethernet). Often, compute nodes have more than one ethernet interface. This procedure describes how to configure them.

Additional ethernet interfaces are configured from the frontend via the Rocks command line. It modifies entries in the networks table on the frontend to add information about an extra interface on a node.

Once you have the information in the networks table, every time you reinstall, the additional NIC will be configured.

Suppose you have a compute node with one configured network (eth0) and one unconfigured network (eth1):

```
# rocks list host interface compute-0-0
SUBNET  IFACE MAC               IP           NETMASK     MODULE NAME        VLAN
private eth0  00:1e:4f:b0:74:ef 10.1.255.254 255.255.0.0 tg3    compute-0-0 ------
------- eth1  00:10:18:31:74:43 ------------ ----------- tg3    ----------- ------
```

We'll configure eth1 with the following network info and associate eth1 with the public subnet:

- Name = fast-0-0
- IP address = 192.168.1.1

```
# rocks set host interface ip compute-0-0 eth1 192.168.1.1
# rocks set host interface name compute-0-0 eth1 fast-0-0
```

Now we'll create a new 'network' and associate it with the new interface:

```
# rocks add network fast 192.168.1.0 255.255.255.0
```

And then we'll check our work:

```
# rocks list network
NETWORK  SUBNET        NETMASK       MTU
private: 10.1.0.0      255.255.0.0   1500
public:  137.110.119.0 255.255.255.0 1500
fast:    192.168.1.0   255.255.255.0 1500
```

Now associate the new network to eth1.

```
# rocks set host interface subnet compute-0-0 eth1 fast
```

The interface eth1 is now configured:

```
# rocks list host interface compute-0-0
SUBNET  IFACE MAC               IP           NETMASK       MODULE NAME        VLAN
private eth0  00:1e:4f:b0:74:ef 10.1.255.254 255.255.0.0   tg3    compute-0-0 ------
fast    eth1  00:10:18:31:74:43 192.168.1.1  255.255.255.0 tg3    fast-0-0    ------
```

After specifying new network settings to a compute-0-0, execute the following command to apply the settings:

```
# rocks sync config
# rocks sync host network compute-0-0
```

If you configuring the interface to another public network, you can set the gateway for the interface with the `rocks add host route` command.

For example, to set the route for the 192.168.1.0 network to 192.168.1.254 for compute-0-0, you'd execute:

```
# rocks add host route compute-0-0 192.168.1.0 192.168.1.254 netmask=255.255.255.0
```

5.5. Compute Node Disk Partitioning

5.5.1. Default Disk Partitioning

The default root partition is 16 GB, the default swap partition is 1 GB, and the default /var partition is 4 GB. The remainder of the root disk is setup as the partition `/state/partition1`.

Only the root disk (the first discovered disk) is partitioned by default. To partition all disks connected to a compute node, see the section Forcing the Default Partitioning Scheme for All Disks on a Compute Node.

Table 5-1. Compute Node -- Default Root Disk Partition

Partition Name	Size
/	16 GB

Partition Name	Size
swap	1 GB
/var	4 GB
/state/partition1	*remainder of root disk*

After the initial installation, all data in the file systems labeled `/state/partitionX` will be preserved over reinstallations.

5.5.2. Customizing Compute Node Disk Partitions

In Rocks, to supply custom partitioning to a node, one must write code in a `<pre>` section and the code must create a file named `/tmp/user_partition_info`. Red Hat kickstart partitioning directives should be placed inside `/tmp/user_partition_info`. This allows users to fully program their cluster nodes' partitions. In the examples below, we'll explore what this means.

5.5.2.1. Single Disk Example

Create a new XML node file that will *replace* the current `partition.xml` XML node file:

```
# cd /export/rocks/install/site-profiles/5.4.3/nodes/
# cp skeleton.xml replace-partition.xml
```

Inside `replace-partition.xml`, add the following section right after the <main> </main> section:

```
<main>
 <!-- kickstart 'main' commands go here -->
</main>

<pre>
echo "clearpart --all --initlabel --drives=hda
part / --size 8000 --ondisk hda
part swap --size 1000 --ondisk hda
part /mydata --size 1 --grow --ondisk hda" &gt; /tmp/user_partition_info
</pre>
```

The above example uses a bash script to populate `/tmp/user_partition_info`. This will set up an 8 GB root partition, a 1 GB swap partition, and the remainder of the drive will be set up as `/mydata`. Additional drives on your compute nodes can be setup in a similar manner by changing the `--ondisk` parameter.

In the above example, the syntax of the data in `/tmp/user_partition_info` follows directly from Red Hat's kickstart. For more information on the `part` keyword, see Red Hat Enterprise Linux 5 Installation Guide : Kickstart Options[1].

User-specified partition mountpoint names (e.g., `/mydata`) cannot be longer than 15 characters.

Then apply this configuration to the distribution by executing:

```
# cd /export/rocks/install
# rocks create distro
```

To reformat compute node `compute-0-0` to your specification above, you'll need to first remove the partition info for `compute-0-0` from the database:

```
# rocks remove host partition compute-0-0
```

Then you'll need to remove the file `.rocks-release` from the first partition of *each disk* on the compute node. Here's an example script:

```
for file in $(mount | awk '{print $3}')
do
 if [ -f $file/.rocks-release ]
 then
  rm -f $file/.rocks-release
 fi
done
```

Save the above script as `/share/apps/nukeit.sh` and then execute:

```
# ssh compute-0-0 'sh /share/apps/nukeit.sh'
```

Then, reinstall the node:

```
# ssh compute-0-0 '/boot/kickstart/cluster-kickstart'
```

5.5.2.2. Software Raid Example

If you would like to use software RAID on your compute nodes, inside `replace-partition.xml` add a section that looks like:

```
<pre>
echo "clearpart --all --initlabel --drives=hda,hdb
part / --size 8000 --ondisk hda
part swap --size 1000 --ondisk hda

part raid.00 --size=10000 --ondisk hda
part raid.01 --size=10000 --ondisk hdb

raid /mydata --level=1 --device=md0 raid.00 raid.01" > /tmp/user_partition_info
</pre>
```

Then apply this configuration to the distribution by executing:

```
# cd /export/rocks/install
```

```
# rocks create distro
```

To reformat compute node `compute-0-0` to your specification above, you'll need to first remove the partition info for `compute-0-0` from the database:

```
# rocks remove host partition compute-0-0
```

Then you'll need to remove the file `.rocks-release` from the first partition of *each disk* on the compute node. Here's an example script:

```
for file in $(mount | awk '{print $3}')
do
 if [ -f $file/.rocks-release ]
 then
  rm -f $file/.rocks-release
 fi
done
```

Save the above script as `/share/apps/nukeit.sh` and then execute:

```
# ssh compute-0-0 'sh /share/apps/nukeit.sh'
```

Then, reinstall the node:

```
# ssh compute-0-0 '/boot/kickstart/cluster-kickstart'
```

5.5.2.3. Programmable Partitioning

Some issues with the above two examples are that 1) you must know the name of the disk device (e.g., hda) and, 2) the partitioning will be applied to all nodes. We can avoid these issues by writing a python program that emits node-specific partitioning directives.

In the next example, we'll use some Rocks partitioning library code to dynamically determine the name of the boot disk.

```
<pre arg="--interpreter /opt/rocks/bin/python">

import rocks_partition

membership = '&membership;'
nodename = '&hostname;'

def doDisk(file, disk):
        file.write('clearpart --all --initlabel --drives=%s\n' % disk)
        file.write('part / --size=6000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part /var --size=2000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part swap --size=2000 --ondisk=%s\n' % disk)
        file.write('part /mydata --size=1 --grow --fstype=ext3 --ondisk=%s\n'
                % disk)

#
# main
```

```
#
p = rocks_partition.RocksPartition()
disks = p.getDisks()

if len(disks) == 1:
 file = open('/tmp/user_partition_info', 'w')
 doDisk(file, disks[0])
 file.close()
</pre>
```

The function `getDisks()` returns a list of discovered disks. In the code sample above, if only one disk is discovered on the node, then the function `doDisk` is called which outputs partitioning directives for a single disk. This code segment will work for nodes with IDE or SCSI controllers. For example, a node with a IDE controller will name its disks `hdX` and a node with SCSI controllers will name its disks `sdX`. But, the code segment above doesn't care how the node names its drives, it only cares if one drive is discovered.

The next example shows how a node can automatically configure a node for software raid when it discovers 2 disks. But, if the node only discovers 1 disk, it will output partitioning info appropriate for a single-disk system.

```
<pre arg="--interpreter /opt/rocks/bin/python">

import rocks_partition

membership = '&membership;'
nodename = '&hostname;'

def doRaid(file, disks):
        file.write('clearpart --all --initlabel --drives=%s\n'
                % ','.join(disks))

        raidparts = []

        for disk in disks:
                if disk == disks[0]:
                        part = 'part / --size=6000 --fstype=ext3 ' + \
                                '--ondisk=%s\n' % disk
                        file.write(part)

                        part = 'part /var --size=2000 --fstype=ext3 ' + \
                                '--ondisk=%s\n' % disk
                        file.write(part)

                part = 'part raid.%s --size=5000 --ondisk=%s\n' % (disk, disk)
                file.write(part)

                raidparts.append('raid.%s' % disk)

        raid = 'raid /bigdisk --fstype=ext3 --device=md0 --level=1 %s\n' \
                % ' '.join(raidparts)
        file.write(raid)

def doDisk(file, disk):
        file.write('clearpart --all --initlabel --drives=%s\n' % disk)
```

```
        file.write('part / --size=6000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part /var --size=2000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part swap --size=2000 --ondisk=%s\n' % disk)
        file.write('part /mydata --size=1 --grow --fstype=ext3 --ondisk=%s\n'
                % disk)

#
# main
#
p = rocks_partition.RocksPartition()
disks = p.getDisks()

file = open('/tmp/user_partition_info', 'w')

if len(disks) == 2:
 doRaid(file, disks)
elif len(disks) == 1:
 doDisk(file, disks[0])

file.close()

</pre>
```

If the node has 2 disks (`if len(disks) == 2:`), then call `doRaid()` to configure a software raid 1 over the 2 disks. If the node has 1 disk then call `doDisk()` and output partitioning directives for a single disk.

In the next example, we show how to output user-specified partitioning info for only one specific node (`compute-0-0`). All other nodes that execute this pre section will get the default Rocks partitioning.

```
<pre arg="--interpreter /opt/rocks/bin/python">

import rocks_partition

membership = '&membership;'
nodename = '&hostname;'

def doRaid(file, disks):
        file.write('clearpart --all --initlabel --drives=%s\n'
                % ','.join(disks))

        raidparts = []

        for disk in disks:
                if disk == disks[0]:
                        part = 'part / --size=6000 --fstype=ext3 ' + \
                                '--ondisk=%s\n' % disk
                        file.write(part)

                        part = 'part /var --size=2000 --fstype=ext3 ' + \
                                '--ondisk=%s\n' % disk
                        file.write(part)

                part = 'part raid.%s --size=5000 --ondisk=%s\n' % (disk, disk)
```

```
                file.write(part)

                raidparts.append('raid.%s' % disk)

        raid = 'raid /bigdisk --fstype=ext3 --device=md0 --level=1 %s\n' \
                % ' '.join(raidparts)
        file.write(raid)

def doDisk(file, disk):
        file.write('clearpart --all --initlabel --drives=%s\n' % disk)
        file.write('part / --size=6000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part /var --size=2000 --fstype=ext3 --ondisk=%s\n' % disk)
        file.write('part swap --size=2000 --ondisk=%s\n' % disk)
        file.write('part /mydata --size=1 --grow --fstype=ext3 --ondisk=%s\n'
                % disk)

#
# main
#
p = rocks_partition.RocksPartition()
disks = p.getDisks()

if nodename in [ 'compute-0-0' ]:
 file = open('/tmp/user_partition_info', 'w')

        if len(disks) == 2:
                doRaid(file, disks)
        elif len(disks) == 1:
                doDisk(file, disks[0])

 file.close()

</pre>
```

5.5.3. Forcing the Default Partitioning Scheme for All Disks on a Compute Node

This procedure describes how to force all the disks connected to a compute node back to the default Rocks partitioning scheme regardless of the current state of the disk drive on the compute node.

The root disk will be partitioned as described in Default Partitioning and all remaining disk drives will have one partition with the name `/state/partition2`, `/state/partition3`, `...`

For example, the following table describes the default partitioning for a compute node with 3 SCSI drives.

Device Name	Mountpoint	Size

Table 5-2. A Compute Node with 3 SCSI Drives

Device Name	Mountpoint	Size
/dev/sda1	/	16 GB
/dev/sda2	swap	1 GB
/dev/sda3	/var	4 GB
/dev/sda4	/state/partition1	*remainder of root disk*
/dev/sdb1	/state/partition2	*size of disk*
/dev/sdc1	/state/partition3	*size of disk*

Create a new XML configuration file that will *replace* the current `partition.xml` configuration file:

```
# cd /export/rocks/install/site-profiles/5.4.3/nodes/
# cp skeleton.xml replace-partition.xml
```

Inside `replace-partition.xml`, add the following section:

```
<pre>
echo "rocks force-default" > /tmp/user_partition_info
</pre>
```

Then apply this configuration to the distribution by executing:

```
# cd /export/rocks/install
# rocks create distro
```

To reformat compute node `compute-0-0` to your specification above, you'll need to first remove the partition info for `compute-0-0` from the database:

```
# rocks remove host partition compute-0-0
```

Then you'll need to remove the file `.rocks-release` from the first partition of *each disk* on the compute node. Here's an example script:

```
for file in $(mount | awk '{print $3}')
do
 if [ -f $file/.rocks-release ]
 then
  rm -f $file/.rocks-release
 fi
done
```

Save the above script as `/share/apps/nukeit.sh` and then execute:

```
# ssh compute-0-0 'sh /share/apps/nukeit.sh'
```

Then, reinstall the node:

```
# ssh compute-0-0 '/boot/kickstart/cluster-kickstart'
```

After you have returned all the compute nodes to the default partitioning scheme, then you'll want to remove `replace-partition.xml` in order to allow Rocks to preserve all non-root partition data.

```
# rm /export/rocks/install/site-profiles/5.4.3/nodes/replace-partition.xml
```

Then apply this update to the distribution by executing:

```
# cd /export/rocks/install
# rocks create distro
```

5.5.4. Forcing Manual Partitioning Scheme on a Compute Node

This procedure describes how to force a compute node to always display the manual partitioning screen during install. This is useful when you want full and explicit control over a node's partitioning.

Create a new XML configuration file that will *replace* the current `partition.xml` configuration file:

```
# cd /export/rocks/install/site-profiles/5.4.3/nodes/
# cp skeleton.xml replace-partition.xml
```

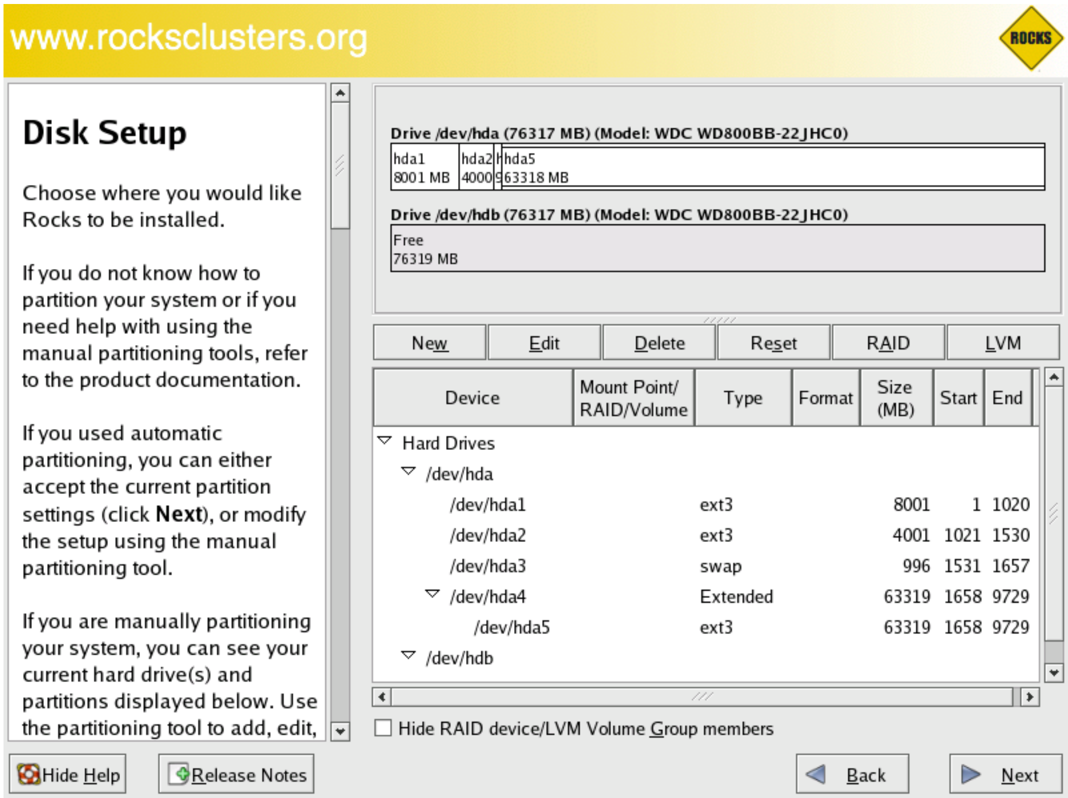
Inside `replace-partition.xml`, add the following section:

```
<pre>
echo "rocks manual" > /tmp/user_partition_info
</pre>
```

Then apply this configuration to the distribution by executing:

```
# cd /export/rocks/install
# rocks create distro
```

The next time you install a compute node, you will see the screen:



To interact with the above screen, from the frontend execute the command:

```
# rocks-console compute-0-0
```

5.6. Creating a Custom Kernel RPM

5.6.1. Creating a Custom Kernel RPM using kernel.org's Source

- On the frontend, check out the Rocks source code. See Access to Rocks Source Code for details.
- Change into the directory:

  ```
  # cd rocks/src/roll/kernel/src/kernel.org
  ```

- Download the kernel source tarball from kernel.org. For example:

  ```
  # wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.24.4.tar.gz
  ```

- Create a kernel "config" file and put it in `config-<version>`

  You can create the config file by using the following procedure:

  ```
  # tar xzf linux-2.6.24.4.tar.gz
  # cd linux-2.6.24.4
  # make menuconfig
  ```

Configure the kernel anyway you need, and after the configuration is over choose to save the configuration in an alternative location. Enter the name of the file as `../config-2.6.24.4`. Finally, exit the configuration and remove the linux-2.6.24.4 directory.

> The *<version>* number must match the version number of the kernel source. For example, if you downloaded `linux-2.6.24.4.tar.gz`, the name of the config file must be `config-2.6.24.4`.

- Update `version.mk`.

  The file `version.mk` has the following contents:

  ```
  NAME            = kernel
  RELEASE         = 1

  VERSION         = 2.6.24.4
  PAE             = 0
  XEN             = 0
  ```

  The `VERSION` value must match that of the linux kernel tarball you downloaded (e.g., 2.6.24.4).

  If you are building a kernel for an i386 system that has more than 4 GB, you'll need to set the PAE (page address extension) flag. This will name the resulting kernel *kernel-PAE*rpm*. If the anaconda installer detects more that 4 GB memory, then it will install the kernel-PAE RPM and not the kernel RPM.

  ```
  PAE             = 1
  ```

  If you are building a kernel that contains Xen support, then set the XEN flag. This will name the resulting kernel *kernel-xen*rpm*.

  ```
  XEN             = 1
  ```

  It is illegal to set both the PAE and XEN flags.

  If you want to build a kernel that the installer will use, then you must unset both the PAE and XEN flags (the default configuration of version.mk). This will name the resulting kernel *kernel*rpm*.

  > If you want to build a custom kernel for both the installer and for the running system and the running system requires either the kernel-PAE or kernel-xen RPM, then you'll have to execute this procedure twice: once to build the installer kernel (PAE = 0 and XEN = 0) and once to build the kernel that will be used on the running system (PAE = 1 or XEN = 1).

- Build the kernel:

  ```
  # make rpm
  ```

- Copy the resulting RPMs into the current distribution:

  ```
  # cp ../../RPMS/<arch>/kernel*rpm /export/rocks/install/contrib/5.4.3/<arch>/RPMS/
  ```

  Where *<arch>* is i386 or x86_64.

- Rebuild the distribution:

```
# cd /export/rocks/install
# rocks create distro
```

- Test the new kernel by reinstalling a compute node:

```
# shoot-node compute-0-0
```

- If the kernel works to your satisfaction, reinstall all the compute nodes that you want to run the new kernel.

5.7. Enabling RSH on Compute Nodes

The default Rocks configuration does not enable rsh commands or login to compute nodes. Instead, Rocks uses ssh as a drop in replacement for rsh. There may be some circumstances where ssh does not have exactly the same semantics of rsh. Further, there may be some users that cannot modify their application to switch from rsh to ssh. If you are one of these users you may wish to enable rsh on your cluster.

> Enabling rsh on your cluster has serious security implicatations. While it is true rsh is limited to the private-side network this does not mean it is as secure as ssh.

Enabling rsh is done by setting an *attribute*. To enable rsh on all compute nodes, execute:

```
# rocks set appliance attr compute rsh true
```

To apply this configuration change to the compute nodes, reinstall all your compute nodes.

If you only want to enable rsh on a specific node (e.g., compute-0-0), execute:

```
# rocks set host attr compute-0-0 rsh true
```

To apply this configuration change to compute-0-0, reinstall compute-0-0.

5.8. Adding a New Appliance Type to the Cluster

This procedure describes how to add a new appliance type to your cluster. This is useful when you want a subset of compute nodes to have specific behavior that is different from the rest of the compute nodes. For example, if you want all the nodes in cabinet 1 to be configured differently from the rest of the compute nodes.

Before you begin, you'll want to be comfortable with the Rocks XML framework that is used to produce a configuration graph. Details on this framework are found in the Reference Guide[2].

First, you'll need to create a new node XML file. This file will contain the configuration scripts and/or packages that will be applied to each of your appliances. Let's call it `my-compute.xml`. This file should be created in the directory `/export/rocks/install/site-profiles/5.4.3/nodes`. Below is the contents of the file:

```
<?xml version="1.0" standalone="no"?>

<kickstart>
```

```
<description>
My specialized compute node
</description>


<changelog>
</changelog>

<post>

<file name="/etc/motd" mode="append">
My Compute Appliance
</file>

</post>

</kickstart>
```

Now, we'll link the above file into the existing XML configuration graph. We'll simply point the above XML node to the existing `compute.xml` node. In object-oriented terms, we are inheriting all the functionality of the compute appliance and then extending it.

To link `my-compute.xml` to `compute.xml`, in the directory `/export/rocks/install/site-profiles/5.4.3/graphs/default`, create the file `my-appliance.xml` and have it contain:

```
<?xml version="1.0" standalone="no"?>

<graph>

<description>
</description>

<changelog>
</changelog>

<edge from="my-compute">
        <to>compute</to>
</edge>

<order gen="kgen" head="TAIL">
        <tail>my-compute</tail>
</order>

</graph>
```

To apply the changes above to the current distribution, execute:

```
# cd /export/rocks/install
# rocks create distro
```

Now we need to add an entry into the Rocks MySQL database. This is accomplished with the rocks command line:

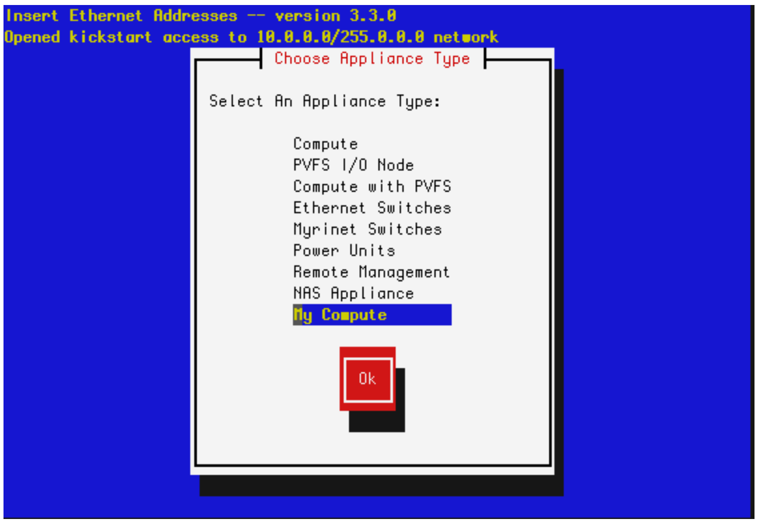
```
# /opt/rocks/bin/rocks add appliance my-compute membership='My Compute' \
```

```
node='my-compute'
```

Now let's retarget an existing compute node. We'll use `insert-ethers` to accomplish this task. First, ask `insert-ethers` to replace `compute-0-0`:

```
# insert-ethers --replace compute-0-0
```

This displays the screen:



Select *My Compute* then hit *Ok*. This removes `compute-0-0` from the database and the next node that asks to be configured (that is, the next node that sends out a DHCP request) will be assigned the name `my-compute-0-0`. To see this in action, instruct `compute-0-0` to reinstall itself:

```
# shoot-node compute-0-0
```

Eventually, you'll see `insert-ethers` report that it discovered `my-compute-0-0`. After the node installs, it will be configured as a *my-appliance*. You can login to the node by executing:

```
# ssh my-compute-0-0
```

Your custom appliance can be applied to any new node in your system by starting `insert-ethers` as instructed above, then by booting a new node in configuration mode (by forcing it to PXE boot or by booting the node with the Kernel/Boot Roll).

5.9. Adding a Device Driver

This section describes how to add a device driver to the installation environment (*initrd.img-5.4.3-<arch>*). This enables the installation environment to use the new driver as well as installing the device driver into the running environment (that is, after the node has installed).

This feature is enabled by ddiskit[3] which is maintained by Jon Masters at Red Hat.

1. Set up a build environment:

```
# cd /export
# hg clone http://fyp.rocksclusters.org/hg/rocks-5.4.3
```

2. Go to the directory which holds the device driver code:

   ```
   # cd /export/rocks-5.4.3/src/roll/kernel/src/rocks-boot/enterprise/5/images/drivers
   ```

3. In this directory, you'll see some example drivers. Let's look at the *e1000* driver:

   ```
   # cd e1000
   ```

4. If you want to supply a new version of the driver, you'll have to download the e1000 source tarball and copy the *.c and *.h files from the tarball to this directory. Make sure all the *.c and *.h files are listed at the top of the *Makefile*:

   ```
   MODULES := e1000

   SOURCES := e1000_main.c e1000_hw.c e1000_ethtool.c e1000_param.c

   HEADERS := e1000.h e1000_hw.h e1000_osdep.h
   ```

5. You'll need to make sure the proper PCI ids are in the file *pcitable*. For example, to test on one of our Dell SC1425's, we added the line:

   ```
   0x8086  0x1076  "e1000" "Intel|82541GI/PI Gigabit Ethernet Controller (rev 05)"
   ```

6. Now we'll need to specify to the device driver building code that the e1000 driver should be built. To do this, edit the file *subdirs*:

   ```
   # cd ..
   # vi subdirs
   ```

7. Change the section from:

   ```
   #
   # put a list of all the driver directories that you'd like to build.
   #
   # for example, to build the 'e1000' driver, uncomment the line below:
   #e1000
   ```

   to:

   ```
   #
   # put a list of all the driver directories that you'd like to build.
   #
   # for example, to build the 'e1000' driver, uncomment the line below:
   e1000
   ```

8. Build the *rocks-boot* package:

   ```
   # cd /export/rocks-5.4.3/src/roll/kernel/src/rocks-boot
   # make rpm
   ```

9. When this completes, copy the binary RPMs into a directory where the distribution building utility (*rocks-dist*) will find and include them:

   ```
   # cp /export/rocks-5.4.3/src/roll/kernel/RPMS/x86_64/rocks-boot* \
    /export/rocks/install/contrib/5.4.3/x86_64/RPMS/
   ```

   If you are building on an i386 system, change the above x86_64 references to i386.

10. Rebuild the distro:

```
# cd /export/rocks/install
# rocks create distro
```

11. Install the newly created *initrd.img-5.4.3-<arch>* and its matching kernel *vmlinuz-5.4.3-<arch>* so PXE booted nodes will get the new device drivers:

```
# cd /export/rocks/install
# rpm -Uvh --force rocks-dist/x86_64/RedHat/RPMS/rocks-boot-5*.rpm
# cp /boot/kickstart/default/initrd.img-5.4.3-<arch> /tftpboot/pxelinux/
# cp /boot/kickstart/default/vmlinuz-5.4.3-<arch> /tftpboot/pxelinux/
```

12. Now PXE boot a node. This node will load your new driver and will install this driver into the running environment.

5.9.1. Adding a New Device Driver (That Isn't One of the Example Drivers)

If the name of your device driver you wish to add is not one of the example device drivers (e.g., ata_piix, cciss, e1000, sk98lin, or tg3), then you'll need to create a new directory and populate it with the appropriate files.

For help on how to populate your new driver directory, see: /export/rocks-5.4.3/src/roll/kernel/src/rocks-boot/enterprise/5/images/drivers/INSTALL.

The rest of the build process follows the same procedure as above starting at step 6 where you'll have to add the name of your new driver to the file *subdirs*.

5.10. Extending DNS

Rocks provides a mechanism to put external hostnames under the DNS control of your cluster. Generally, external hosts have names served by site-wide DNS servers. However if there is no external DNS server available, you may want to use your frontend's DNS server to handle the name to IP mappings for certain non-cluster nodes.

Since the DNS configuration file is automatically generated by Rocks, you cannot add static configuration to the standard zone files in /var/named. Instead, you need to put local name mappings in the file:

`/var/named/<networkname>.domain.local`

And reverse mappings (IP to name) in:

`/var/named/reverse.<networkname>.domain.<networkprefix>.local`

Where *networkname* for a default Rocks installation is *private* and *networkprefix* for a default Rocks installation is *1.10*. That is, the above file names would be:

For forward mapping: `/var/named/private.domain.local` For reverse mapping: `/var/named/reverse.private.domain.1.10.local`

To add the mappings in the above *.local* files to the running configuration, execute:

```
# rocks sync config
```

These files are in the BIND configuration format, just like the standard `private.domain` and `reverse.private.domain` files.

Your external hosts will have names in the .local cluster domain.

Errors in your local DNS files will cause the entire local cluster domain naming to fail. Proceed with caution.

5.11. Changing the Root password

Starting with Rocks v5.4.3, we have attempted to improve the security of the Rocks system by making changes to the way some of the important passwords are generated and stored.

To change the root password on the frontend, you can now run the **passwd** command.

To change the root password on the client nodes, you can use the rocks secure attribute infrastructure.

Example 5-1. Change the root password of a single compute host

```
# rocks set host sec_attr compute-0-0 attr=root_pw
  Enter root_pw:
Confirm root_pw:
# rocks list host sec_attr compute
HOST          ATTR    VALUE                              ENC
compute-0-0: root_pw $1$G55uZRrO$5MI7Nv14U5yWfKAlauqPE0 crypt
# rocks sync host sec_attr compute-0-0
```

For more information about changes to the rocks security infrastructure, refer to the Rocks Password Infrastructure section of this usersguide

The default rocks password policy is to set the root password of the frontend to be the password supplied during the installation. For client nodes, the root password will be generated at random during installation. Unless there is compelling reason to do so, (such as giving a user root access to a compute node) the Rocks group advices against changing the root password on the client nodes.

Notes

1. http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/en-US/RHEL510/Installation_Guide/s1-kickstart2-options.html
2. http://www.rocksclusters.org/rocks-documentation/reference-guide/4.3/

3. http://dup.et.redhat.com/ddiskit/

Chapter 6. Community Resources

6.1. Access to Rocks Source Code

The source code for Rocks is available from a public read-only Mercurial[1] repository. As of Rocks 5.0 Mercurial (hg) is installed on your frontend machine, for older version of Rocks (or non-Rocks platforms) you will need to install the Mercurial package yourself. Binary packages for Mercurial are available here[2].

Mercurial is a distributed source control system. A very good explanation on how this differs from CVS is available on the Mercurial site[3]. Fundamentally, distributed source control turns your checked out source code from a read-only copy into a repository and then grants read-write access to others based on your checked out repository. In the context of Rocks this means individual Roll developers can independently control who has access to their code.

To check out the current source code you will first need to clone the master Rocks read-only repository as follows. This operation will take several minutes (if not hours) and unlike a CVS checkout it may appear to hang, be patient.

```
$ hg clone http://fyp.rocksclusters.org/hg/rocks-5.4.3
destination directory: rocks-5.4.3
real URL is http://fyp.rocksclusters.org/hg/rocks-5.4.3/
requesting all changes
adding changesets
adding manifests
adding file changes
added 1 changesets with 2815 changes to 2815 files
2815 files updated, 0 files merged, 0 files removed, 0 files unresolved
```

6.2. All Past Rocks Releases

All past Rocks releases can be found at `ftp.rocksclusters.org:/pub/rocks`.

6.3. Email Discussion List

The primary mode for support for Rocks is through our email discussion list.

Click here[4] to subscribe.

The archives[5] hold messages from the Rocks discussion list dating back to June 2002.

6.4. Office Hours

Another mode in which to get support for Rocks is through "Office Hours". The idea is to have a set time in which members of the Rocks community can have a one-on-one live conversation with a Rocks developer.

Office Hours is held every Tuesday from 11:30 AM to 12:30 PM Pacific Time. If that time slot is inconvienient for you, send a note to officehours@rocksclusters.org and we'll try to set up an appointment.

To contact a Rocks developer during Office Hours, you can:

- Make an audio or video conference call with Skype to the username "rocksclusters".
- Call the number 858.866-9422 (country code 1).

Notes

1. http://www.selenic.com/mercurial/wiki/
2. http://www.selenic.com/mercurial/wiki/index.cgi/BinaryPackages
3. http://www.selenic.com/mercurial/wiki/index.cgi/CvsConcepts
4. https://lists.sdsc.edu/mailman/listinfo/npaci-rocks-discussion
5. https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/

Chapter 7. Administration Examples

7.1. Introduction to the Rocks Command Line

In Rocks 4.3 the Rocks command line was introduced to provide a more uniform interface to the underlying structures used to control system configuration and behaviour. Wherever possible, Rocks uses a SQL database (MySQL currently) to hold information about nodes, partitioning information, boot parameters and other information. Based on information in the database, various configuration files are rewritten. The re-generation of configuration files occurs everytime a node is added or deleted from the cluster. The re-generation of configuration files can also be forced. A large fraction of Rocks commands manipulate data held in the configuration database. In general, the process of changing configuration is a two-step process:

1. Use rocks commands to change configuration in the database (e.g. `rocks set host`)
2. Rewrite configuration files using `rocks sync config`

It should be noted that step 1 above is usually called several times to update in the database and then step 2 is called to write individual confiuration files in the format that the native OS tools understand.

Rocks commands have arguments and parameters. Parameters are of the form "param=<value>" and may appear anywhere. Arguments must appear in the order defined by the command. To get help on any rocks command type "help" for the argument to the command. For example `rocks set host interface ip help`

7.2. Boot Order and PXE First

Prior to Rocks 4.3, the BIOS-defined boot order of a compute node *required* that a network boot (known as PXE) come after local hard disk. In particular the boot order in BIOS would be set as

```
1. CDROM
2. Hard Disk
3. On-board Network Device (PXE)
```

A user would have to intercept the boot sequence (often by hitting the F12 key on an attached keyboard) to force a network boot. Rocks also provided a small utility on each node (`/boot/kickstart/cluster-kickstart-pxe`) that would manipulate the two-bytes on the local hard disk to force BIOS to bypass booting from the local disk and try the next device on the boot list. When the boot order was set as above, the node would PXE boot and therefore re-install.

The logic for this structure was that a frontend did not need to know the state of node (whether it had failed and should be reinstalled or had some other intermediate state). Also it is not required that a frontend be up for a node to reboot itself. Another practical issue arises for PXE booting large clusters. Since the PXE client is in NIC firmware, no assumptions about timeouts, retries or other elements that figure into robustness could be made. Large cluster reinstalls (or reboots) for a kernel that comes over PXE would often result in hung nodes because of the low level of robustness of TFTP (the underlying protocol used to transfer initial kernel and ramdisk image for nodes booting over the network). For wholesale re-installation of large clusters, PXE does not scale well. For this, Rocks provides the installation kernel and initial ramdisk image on the local hard drive. The commmand `/boot/kickstart/cluster-kickstart` run on a local node will cause that node to re-install itself by using a local (hard disk) copy of the installation kernel and initial ramdisk.

> The above boot order and behaviour continues to be supported in Rocks 4.3. That is, existing rocks clusters can be upgraded without requiring the cluster owner to change any BIOS settings.

7.3. Support for PXE First

Rocks supports a network device first (or PXE first) BIOS-defined boot order. It is now *recommended* that a network boot (known as PXE) come before local hard disk. In particular the boot order in BIOS should be set as

```
1. CDROM
2. On-board Network Device (PXE)
3. Hard Disk
```

The default PXE "action" is to simply pass to the next device down on the BIOS boot list. In the usual case, this is to the local hard disk. Most of the time decision to boot or reinstall is still left to the local node and the frontend does not need to know which state the node desires. If booting into re-installation (e.g. the node either did not shut down properly, or `/boot/kickstart/cluster-kickstart` was called locally) that will proceed as expected. However, it is possible to change this action on a per-node basis.

7.4. Forcing a Re-install at Next PXE Boot

Starting with Rocks 4.3, the frontend must be configured to tell a node to re-install at the next PXE boot. This action is controllable on a per-node basis. At the end of successful installation, the node requests the frontend to set its PXE boot to *os*. To re-install a node using PXE (e.g. compute-0-0), then do the following:

```
# rocks set host boot compute-0-0 action=install
# ssh compute-0-0 "shutdown -r now"
```

> If the boot order has not been set to PXE first, you can force a PXE boot with the local keyboard, or by calling `/boot/kickstart/cluster-kickstart-pxe` on the local node.

7.5. Inspecting and Changing PXE Behaviour

There are two parts to the Rocks database for modifying PXE behaviour: *boot* and *bootaction*. The "boot" part determines which logical action should be performed. The two common actions are "os" and "install." The second table is the bootaction table. This associates a logical action with a specific TFTP configuration.

It is possible to have commands affect all nodes. In this case use '%' as the host wildcard. For example `rocks set host boot % action=install` will cause ALL nodes to reinstall the next time they boot.

For commands that take lists of hosts, it is possible to use an appliance type for the host(s) argument. `rocks list appliance` are the list of valid appliance types. To set the boot action of all compute appliances to be *install*, use `rocks set host boot compute action=install`.

The following illustrates how to inspect the current action of nodes and then the specifics of each action.

```
# rocks list host boot
HOST          ACTION
vizzy:        os
compute-0-0: os
compute-0-1: os
compute-1-0: os
compute-1-1: install
compute-2-0: os
compute-2-1: os


# rocks list bootaction output-col=action,kernel
ACTION                KERNEL
install:              vmlinuz-5.2-i386
install headless:     vmlinuz-5.2-i386
memtest:              kernel memtest
os:                   localboot 0
pxeflash:             kernel memdisk bigraw
rescue:               vmlinuz-5.2-i386
```

In the above, all nodes are set to boot the "os", except for node compute-1-1. That node will call the boot action named "install". In the case the TFTP configuration file contain the details arguments of the listed in the install action. The command `rocks list bootaction` shows the details of each logical action.

7.5.1. Changing a logical PXE action

It is possible to override the details of a logical action on a per-node basis. Suppose that we wanted to make the logical action of "install" for compute-1-1 to be headless and to set a flag `acpi=off`. Then the following will accomplish this:

```
# rocks add bootaction action="install headless noacpi" kernel="vmlinuz-5.2-i386" ramdisk="initrd.im
args="ks ramdisk_size=150000 lang= devfs=nomount pxe kssendmac selinux=0 noipv6 headless vnc acpi=of

# rocks set host installaction compute-1-1 action="install headless noacpi"
```

To inspect that the change is indeed specific to just compute-1-1, then do the following

```
# rocks list host compute-1-1
MEMBERSHIP   CPUS RACK RANK RUNACTION INSTALLACTION
Compute      2    0    0    os        install headless noacpi

# rocks list host compute-1-0
MEMBERSHIP   CPUS RACK RANK RUNACTION INSTALLACTION
Compute      8    0    1    os        install
```

In the above, compute-1-1 has a specific override for its install action, where compute-1-0 still retains the default install action.

7.5.2. Running Memtest86

It is often useful to run the memory testing tool memtest86+ [1] to determine if memory is valid. The straightforward way to accomplish this is to apply the following procedure (in our example case for host compute-1-1)

1. `# rocks set host runaction compute-1-1 action=memtest`

   `# rocks set host boot compute-1-1 action=os`
2. Boot node compute-1-1 by power cycle or other means.
3. After compute-1-1 has successfully started the diagnostic, reset the runaction parameter:

   `# rocks set host runaction compute-1-1 action=os`

7.6. Working with and Modifying Network Configuration

The Rocks database holds information that has been discovered about a host and in particular records network interface information including MAC addresses and local device modules. The Rocks command line has several tools to inspect and update entries in the database.

For the following discussion, a compute node with a public interface will be used for illustation. To list the ethernet interfaces do the following:

```
# rocks list host interface compute-0-0

SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME                   VLAN OPTI
private eth0  00:0e:0c:a7:57:d7 10.1.255.254   255.255.0.0   ------ compute-0-0            ---- ----
public  eth1  00:19:b9:21:b8:b6 137.110.119.93 255.255.255.0 ------ kong.rocksclusters.org ---- ----
```

Suppose that it is desired to swap these two interfaces. That is, we would want to associate the private network with eth1 and the public network with eth0. The following will change the information only in the database.

```
# rocks swap host interface compute-0-0 ifaces=eth0,eth1

# rocks list host interface compute-0-0
SUBNET  IFACE MAC               IP             NETMASK       MODULE NAME                    VLAN OPTI
public  eth0  00:0e:0c:a7:57:d7 137.110.119.93 255.255.255.0 ------ kong.rocksclusters.org ---- ----
private eth1  00:19:b9:21:b8:b6 10.1.255.254   255.255.0.0   ------ compute-0-0             ---- ----
```

We see that eth0 is now associated with the public network, and eth1 is associated with the private network.

To apply the changes to compute-0-0, execute:

```
# rocks sync config
# rocks sync host network compute-0-0
```

7.7. Reinstall All Compute Nodes with SGE

This section describes how to reinstall all the nodes under the control of SGE.

As root, execute:

```
# /opt/gridengine/examples/jobs/sge-reinstall.sh
```

This will produce output like:

```
Your job 1 ("reboot.qsub") has been submitted
Set compute-0-0.local for Reinstallation
Your job 2 ("reboot.qsub") has been submitted
Set compute-0-1.local for Reinstallation
Your job 3 ("reboot.qsub") has been submitted
Set compute-0-2.local for Reinstallation
```

For each compute node, the script submits a high-priority reinstall job. The reinstall jobs will move ahead of the currently queued jobs (running jobs will not be disturbed). When a busy compute node becomes free, the next job submitted to it will be the reinstall job. All remaining jobs will remain queued until the node completes its installation, after which SGE will recognize the node as up and then submit a previously queued job to it.

Notes

1. http://www.memtest.org

Chapter 8. Advanced Tasks

8.1. Managing the Firewall on the Cluster Nodes

The firewall for all cluster nodes (including the frontend) is managed with the Rocks command line. As of Rocks 5.4.3, all firewall rules must have a name associated. The name is used as the handle for deleting and redefined rules. For example, a rule named ’MYRULE’ can be defined globally, but a particular host can redefine ’MYRULE’ to do something else. As of Rocks 5.4.3 rules are associated by category and single ’add firewall’ command handles global, os, appliance, and host categories.

8.1.1. Categories. Subtle Syntax

In Rocks 5.4.3 and newer releases, the concept of "categories" is used by the database. Prior to Rocks 5.4.3, there were several different add firewall commands. Namely,

1. rocks add firewall
2. rocks add os firewall
3. rocks add appliance firewall
4. rocks add host firewall

Logically, these were defined to add a firewall command to one of [global, os , appliance, host]. When firewall rules were resolved from the database, rules would be inherited from global, then os, then appliance, and finally host-specific rules.

As of 5.4.3, these have been replaced by a single command that looks like

```
# rocks add firewall category=index ...
```

Here categories are [global, os, appliance, host]. This means that adding firewall rule to a particular host (say, compute-0-0) becomes

```
# rocks add firewall host=compute-0-0 ...
```

This rule applies only to the compute-0-0 host. To add a rule to the Compute appliance becomes:

```
# rocks add firewall appliance=Compute ...
```

The appliance rule applies to all Compute appliances.

8.1.1.1. Listing Firewall Rules. Resolved Rules.

There are two commands for listing firewall rules for a host and they differ in subtle, but important, ways. Again we'll use compute-0-0 as an example.

```
# rocks list firewall host=compute-0-0
# rocks list host firewall compute-0-0
```

The first command lists all firewall rules that are specific to host compute-0-0 only. By default, this is the empty set. The second command list all *inherited* commmands and is, by default not empty. For example,

```
# rocks list firewall host=compute-0-0
#
```

and

```
# rocks list host firewall compute-0-0 maxwidth=15
# rocks list host firewall compute-0-0-0 maxwidth=15
RULENAME             SERVICE PROTOCOL CHAIN  ACTION NETWORK OUTPUT-NETWORK FLAGS           COMMENT CA
A20-ALL-PRIVATE      all     all      ACCEPT INPUT  private -------------- --------------- ------- gl
A20-SSH-PUBLIC       ssh     tcp      ACCEPT INPUT  public  -------------- -m state --stat ------- gl
A30-RELATED-PUBLIC   all     all      ACCEPT INPUT  public  -------------- -m state --stat ------- gl
R900-PRIVILEGED-TCP  all     tcp      REJECT INPUT  public  -------------- --dport 0:1023  ------- gl
R900-PRIVILEGED-UDP  all     udp      REJECT INPUT  public  -------------- --dport 0:1023  ------- gl
```

Notice that in the second form, the source of the firewall (global, in this example) is listed so that an administrator can see where a rule is defined. One should also notice that list host firewall will output all possible firewall rules that *might* apply. For example, the above has rules defined for both the "public" and "private" networks in the global scope. If compute-0-0 does not have an interface on the public network, then those rules will not appear in the final firewall configuration.

The list commands (like all list commands in Rocks) are intended for human readability. While the report commmands are machine readable. The actual firewall rules written for compute-0-0 would are generated with `rocks report host firewall`. The report command resolves that actual interface for the named network. For example:

```
#rocks report host firewall compute-0-0
<file name="/etc/sysconfig/iptables" perms="500">
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
#  A20-ALL-PRIVATE (global) :
-A INPUT -i eth0 -j ACCEPT
COMMIT
</file>
```

A comment is put into this file for each applicable firewall rule. Notice that compute-0-0 has only a private network, so that all firewall rules bound to a public network are ignored. `rocks report host firewall` is used internally during node build and by `rocks sync host firewall`

8.1.2. Adding Rules. Ordering Rules. Examples

Firewall concepts are taken from linux iptables. The defined chains are INPUT, OUTPUT, FORWARD, ACCEPT, REJECT, DROP. The most common is to define how to match a packet on the INPUT chain and then either ACCEPT, REJECT, or DROP.

Overall order of rule definition matters and this is where the rulename is important. Suppose that on the frontend, we want to accept ftp connections from all hosts, instead of the rocks default. First lets list the rules for a frontend (COMMENT field has been edited out)

```
RULENAME             SERVICE PROTOCOL CHAIN      ACTION      NETWORK OUTPUT-NETWORK FLAGS
A10-REJECT-411-TCP   all     tcp      REJECT     INPUT       private -------------- --dport 372 --sp
A10-REJECT-411-UDP   all     udp      REJECT     INPUT       private -------------- --dport 372 --sp
A20-ALL-PRIVATE      all     all      ACCEPT     INPUT       private -------------- ----------------
A20-SSH-PUBLIC       ssh     tcp      ACCEPT     INPUT       public  -------------- -m state --state
A30-RELATED-PUBLIC   all     all      ACCEPT     INPUT       public  -------------- -m state --state
A40-HTTPS-PUBLIC-LAN https   tcp      ACCEPT     INPUT       public  -------------- -m state --state
A40-WWW-PUBLIC_LAN   www     tcp      ACCEPT     INPUT       public  -------------- -m state --state
A50-FORWARD-RELATED  all     all      ACCEPT     FORWARD     public  private        -m state --state
A60-FORWARD          all     all      ACCEPT     FORWARD     private -------------- ----------------
MASQUERADE           nat     all      MASQUERADE POSTROUTING ------- public         ----------------
R10-GANGLIA-UDP      8649    udp      REJECT     INPUT       ------- -------------- ----------------
R20-MYSQL-TCP        3306    tcp      REJECT     INPUT       ------- -------------- ----------------
R30-FOUNDATION-MYSQL 40000   tcp      REJECT     INPUT       ------- -------------- ----------------
R900-PRIVILEGED-TCP  all     tcp      REJECT     INPUT       public  -------------- --dport 0:1023
R900-PRIVILEGED-UDP  all     udp      REJECT     INPUT       public  -------------- --dport 0:1023
```

The R900-PRIVILEGED* rules are the last ones interpreted by iptables. They instruct iptables to reject all packets destined for privileged ports on the public network `(--dport 0:1023)`. To accept ftp traffic (port 21) we need to add rules for both UDP and TCP traffic that are named *alphabetically* before the R900 rules. It is the rulename that determines order. Typically ACCEPT rules are labeled A<nn>- and REJECT rules are labeled R<nn>-. But that is only convention for Rocks system rules. Any name is valid.

```
# rocks add firewall host=frontend network=public protocol=tcp service=ftp  chain=INPUT action=ACCEP
# rocks add firewall host=frontend network=public protocol=udp service=ftp  chain=INPUT action=ACCEP
```

It is sometimes simpler on the screen to look at the actual machine report via `rocks report host firewall`

```
# rocks report host firewall frontend
<file name="/etc/sysconfig/iptables" perms="500">
*nat
#  MASQUERADE (host) :
-A POSTROUTING -o eth1 -j MASQUERADE
```

```
COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
#  A10-REJECT-411-TCP (host) :
-A INPUT -i eth0 -p tcp --dport 372 --sport 1024:65535 -j REJECT
#  A10-REJECT-411-UDP (host) :
-A INPUT -i eth0 -p udp --dport 372 --sport 1024:65535 -j REJECT
#  A100-PUBLIC-FTP-UDP (host) :
-A INPUT -i eth1 -p udp --dport ftp -j ACCEPT
#  A20-ALL-PRIVATE (global) :
-A INPUT -i eth0 -j ACCEPT
#  A20-SSH-PUBLIC (global) :
-A INPUT -i eth1 -p tcp --dport ssh -m state --state NEW -j ACCEPT
#  A30-RELATED-PUBLIC (global) :
-A INPUT -i eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
#  A40-HTTPS-PUBLIC-LAN (host) :
-A INPUT -i eth1 -p tcp --dport https -m state --state NEW --source &Kickstart_PublicNetwork;/&Kicks
#  A40-WWW-PUBLIC_LAN (host) :
-A INPUT -i eth1 -p tcp --dport www -m state --state NEW --source &Kickstart_PublicNetwork;/&Kicksta
#  A50-FORWARD-RELATED (host) :
-A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
#  A60-FORWARD (host) :
-A FORWARD -i eth0 -j ACCEPT
#  A90-PUBLIC-FTP-TCP (host) :
-A INPUT -i eth1 -p tcp --dport ftp -j ACCEPT
#  R10-GANGLIA-UDP (host) : block ganglia traffic from non-private interfaces
-A INPUT -p udp --dport 8649 -j REJECT
#  R20-MYSQL-TCP (host) : block mysql traffic from non-private interfaces
-A INPUT -p tcp --dport 3306 -j REJECT
#  R30-FOUNDATION-MYSQL (host) : block foundation mysql traffic from non-private interfaces
-A INPUT -p tcp --dport 40000 -j REJECT
#  R900-PRIVILEGED-TCP (global) :
-A INPUT -i eth1 -p tcp --dport 0:1023 -j REJECT
#  R900-PRIVILEGED-UDP (global) :
-A INPUT -i eth1 -p udp --dport 0:1023 -j REJECT
```

There are four things to notice. First, the ordering is alphabetical not numerical. A100 is after A10-R but before A20. A90 is after A60. Second, both of the accept rules for ftp come before the reject rules. Third, the public network has been resolved for this host to be eth1. See `rocks list host interface frontend` . And, fourth, each lists where it comes from.

Now apply the configuration to the host:

```
# rocks sync host firewall frontend
```

The host will now accept ftp traffic on its public interface.

8.1.2.1. How to affect a particular appliance

Suppose we wanted to close ftp on *all* networks for compute appliances (even private). First, let's look at the default rules for a compute node.

```
rocks list host firewall compute-0-0
RULENAME             SERVICE PROTOCOL CHAIN  ACTION NETWORK OUTPUT-NETWORK FLAGS                    C
A20-ALL-PRIVATE      all     all      ACCEPT INPUT  private -------------- ------------------------ -
A20-SSH-PUBLIC       ssh     tcp      ACCEPT INPUT  public  -------------- -m state --state NEW     -
A30-RELATED-PUBLIC   all     all      ACCEPT INPUT  public  -------------- -m state --state RELATED -
R900-PRIVILEGED-TCP  all     tcp      REJECT INPUT  public  -------------- --dport 0:1023           -
R900-PRIVILEGED-UDP  all     udp      REJECT INPUT  public  -------------- --dport 0:1023           -
```

Here you can use the "appliance=<appliance-name>" argument for adding a firewall. Use `rocks list appliance` to get the valid names. Notice that by default all traffic is accepted on the compute node on the private network. If we want our REJECT rule to properly apply to all network interfaces, it must come before the A20-ALL-PRIVATE rule.

```
# rocks add firewall appliance=compute protocol=tcp service=ftp network=all chain=INPUT action=REJEC
# rocks add firewall appliance=compute protocol=udp service=ftp network=all chain=INPUT action=REJEC
```

This now rejects. FTP traffic. You can check the actual firewall rule for a particular compute host.

```
rocks report host firewall compute-0-0
<file name="/etc/sysconfig/iptables" perms="500">
*filter
:INPUT ACCEPT [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
#  A19-REJECT-FTP-TCP (appliance) :
-A INPUT -p tcp --dport ftp -j REJECT
#  A19-REJECT-FTP-UDP (appliance) :
-A INPUT -p udp --dport ftp -j REJECT
#  A20-ALL-PRIVATE (global) :
-A INPUT -i eth0 -j ACCEPT
COMMIT
</file>
```

Now apply the firewall on all compute nodes.

```
# rocks sync host firewall compute
```

Firewall is now set to reject FTP on all networks for all compute nodes.

8.2. Flashing BIOS on Compute Nodes Using PXE

This procedure describes how to flash BIOS on a client node (e.g., compute node or viz tile node) by using the Rocks command line and PXE.

- First, you must add the BIOS flashing files (e.g., a DOS-based flash utility, .bin file or an autoexec.bat) to the directory `/opt/pxeflash/addon`.

  As an example, to flash a Dell Dimension E521 desktop, we executed:

  ```
  # cd /opt/pxeflash/addon
  # wget http://ftp.us.dell.com/bios/DME521-010111.EXE
  ```

- To add the BIOS flashing file to a bootable floppy image, execute:

  ```
  # cd /opt/pxeflash
  # make build
  ```

  This will incorporate all files in the `/opt/pxeflash/addon` directory into a floppy image named `pxeflash-FDSTD.288`.

- Now copy the floppy image into `/tftpboot/pxelinux/pxeflash`, by executing:

  ```
  # make install
  ```

- Set a client node to PXE boot your flash image:

  ```
  # rocks set host runaction <hostname> action=pxeflash
  # rocks set host boot <hostname> action=run
  ```

- PXE boot the client node. After the client boots, it will display a DOS prompt.
- On the client, execute the BIOS flash program.

  In our example, we executed the program `DME521-010111.EXE`.

- When the BIOS flash program completes and before you reboot the client, on the frontend, reset the PXE action so when the client reboots, it will boot the OS from local disk:

  ```
  # rocks set host runaction <hostname> action=os
  # rocks set host boot <hostname> action=os
  ```

- When the flash process completes, reboot the client node.

Important Acknowledgement: We are grateful for the contribution of Jeremy Lawrence for describing the basic approach in his Rocks discussion list posting https://lists.sdsc.edu/pipermail/npaci-rocks-discussion/2005-June/012092.html.

8.3. Adding a Login Appliance to Your Cluster

A *login appliance* is a host that users use to build, launch and monitor their application(s). This host is a subset of a frontend appliance (a frontend runs several other services that are used to install and maintain all the nodes in the system).

First, we need to perform the initial installation of the login appliance. On the frontend, execute:

```
# insert-ethers
```

Select "Login" as the appliance type.

Then PXE boot the host that you want to be a login appliance.

Login hosts are generally hosts that also have a public network interface so users can directly access the login host. After the login appliance installs, to configure the public network interface on the login host, see Adding a public IP address to the second ethernet adapter on a compute node.

8.3.1. Making any Host a Submission and/or Execution Host

If you've installed a batch queueing system (e.g., by using the SGE Roll or Torque Roll), you can change any host in your system to be a submission host (a host where users can submit jobs) and/or an execution host (the queueing system schedules jobs on execution hosts).

By default, a login host is a submission host, but not an execution host. By default, a compute node is an execution host, but not a submission host.

To make a host a submission host (e.g., tile-0-0), on the frontend execute:

```
# rocks set host attr tile-0-0 submit_host true
```

Then reinstall tile-0-0.

To make a host an execution host (e.g., tile-0-0), on the frontend execute:

```
# rocks set host attr tile-0-0 exec_host true
```

Then reinstall tile-0-0.

8.4. Channel Bonding Interfaces

Channel bonding enables two or more network interfaces to act as one. The following procedure describes how to channel bond interfaces in Rocks.

We'll use the example of channel bonding two interfaces for compute-0-1. First, let's list the current configuration of the interfaces for compute-0-1:

```
# rocks list host interface compute-0-1
SUBNET  IFACE MAC               IP           NETMASK     MODULE NAME        VLAN OPTIONS
private eth0  00:1e:4f:b0:74:ef 10.1.255.253 255.255.0.0 tg3    compute-0-1 ---- -------
------- eth1  00:10:18:31:74:43 ------------ ----------- tg3    ----------- ---- -------
```

Here's what we'll do:

- Bond eth0 and eth1.
- Name the bonded channel "bond0".
- Give bond0 the IP address 10.1.255.253.
- Associate bond0 with the private network.

```
# rocks add host bonded compute-0-1 channel=bond0 interfaces=eth0,eth1 ip=10.1.255.253 network=priva
```

The above command changes the interface configuraion for compute-0-1 in the following way:

```
# rocks list host interface compute-0-1
SUBNET  IFACE MAC               IP           NETMASK     MODULE  NAME        VLAN OPTIONS CHANNEL
private bond0 ----------------- 10.1.255.253 255.255.0.0 bonding compute-0-1 ---- ------- -------
------- eth0  00:1e:4f:b0:74:ef ------------ ----------- tg3     ----------- ---- ------- bond0
------- eth1  00:10:18:31:74:43 ------------ ----------- tg3     ----------- ---- ------- bond0
```

To apply the configuration, execute:

```
# rocks sync config
# rocks sync host network compute-0-1
```

8.4.1. Adding Parameters to the Kernel Bonding Module

There are several options that can be added to the kernel bonding module (see RedHat's documentation The Channel Bonding Module [1] for a full description of all the options).

To set options "miimon=100" and "mode=balance-rr" for the kernel bonding module associated with bond0, execute:

```
# rocks set host interface options compute-0-1 bond0 options="miimon=100 mode=balance-rr"
```

The "options" value (e.g., "miimon=100 mode=balance-rr") must be a space-separated list.

And now we see:

```
SUBNET  IFACE MAC               IP           NETMASK     MODULE  NAME        VLAN OPTIONS
private bond0 ----------------- 10.1.255.253 255.255.0.0 bonding compute-0-1 ---- miimon=100 mode=ba
------- eth0  00:1e:4f:b0:74:ef ------------ ----------- tg3     ----------- ---- ------------------
------- eth1  00:10:18:31:74:43 ------------ ----------- tg3     ----------- ---- ------------------
```

To apply the options to the bond0 interface, execute:

```
# rocks sync host network compute-0-1
```

8.5. Frontend Central Server

A Central Server is a Rocks Frontend node that can kickstart other frontends and provide rolls over the network, in a process called WAN kickstart. All Rocks frontends have the ability to act as central servers.

The standard Rocks distribution (located under /export/rocks/install) contains a distribution suitable for WAN kickstart. The only steps you must take is to open "www" and "https" access on your frontend for the purpose of RPM package transfer. See Enable WWW access.

8.5.1. Adding Rolls to Serve from a Central

You may wish to serve rolls from your central server that you have not installed on your central server when the central server was initially installed. All frontends will serve the rolls they were built with to client frontends, but often it is advantageous to serve other rolls as well.

First, you must download the Roll ISO image to your central. Then, execute:

```
# rocks add roll <rollname>*.iso
```

8.6. Cross Kickstarting

Rocks supports heterogeneous clusters that contain nodes of different hardware architectures with a process called cross-kickstarting. To support an architecture different than its own, a frontend needs to expand its local distribution with additional packages. This section describes how to install distributions for other architectures on your frontend.

Start with a frontend node, as described by Install Frontend, or Upgrade Frontend. Follow the instructions below for every desired architecture.

8.6.1. Cross Kickstarting x86_64 Compute Nodes with a i386 Frontend

For this example, we assume the frontend is an i386 (32-bit) and the compute nodes are x86_64 CPUs (64-bit).

1. Download the x86_64 Rocks rolls. At a minimum, you'll need to download the x86_64 versions of the Base, Kernel, Web Server and OS disk 1 and OS disk 2 rolls.

   For each roll, add it to the frontend's roll repository:

   ```
   # rocks add roll <roll-name>.iso
   # rocks enable roll <roll-name>
   ```

2. Rebuild your distribution:

   ```
   # cd /export/rocks/install
   # rocks create distro
   # rocks create distro arch=x86_64
   ```

3. To prepare the frontend to cross-kickstart compute nodes via PXE, you first need to add the `rocks-boot` package from the x86_64 architecture:

   ```
   # rpm -i --force --ignorearch /export/rocks/install/rocks-dist/x86_64/RedHat/RPMS/rocks-boot-5.4.
   # cp /boot/kickstart/default/vmlinuz-5.4.3-x86_64 /tftpboot/pxelinux
   # cp /boot/kickstart/default/initrd.img-5.4.3-x86_64 /tftpboot/pxelinux
   ```

4. Then you need to add a new `bootaction`:

   ```
   # rocks add bootaction action="install x86_64" kernel="vmlinuz-5.4.3-x86_64" \
     ramdisk="initrd.img-5.4.3-x86_64" \
     args="ks ramdisk_size=150000 lang= devfs=nomount pxe kssendmac selinux=0 noipv6"
   ```

5. The next step requires you to PXE boot the cross-kickstarted node twice: the first is to discover the node, and a second time to install it.

   On the frontend, run insert-ethers to discover the compute node:

   ```
   # insert-ethers
   ```

   Then, PXE boot the compute node. After the compute node is discovered by insert-ethers, kill insert-ethers (by hitting the F9 key), then turn off the compute node.

   Now instruct the node to use the "install x86_64" profile when it PXE boots (the command below assumes the compute node's name is compute-0-0):

   ```
   # rocks set host installaction compute-0-0 action="install x86_64"
   ```

   Now PXE boot the compute node again. The compute node will now install itself with the x86_64 distribution.

8.6.2. Cross Kickstarting i386 Compute Nodes with a x86_64 Frontend

For this example, we assume the frontend is a x86_64 (64-bit) and the compute nodes are i386 CPUs (32-bit).

1. Download the i386 Rocks rolls. At a minimum, you'll need to download the i386 versions of the Base, Kernel, Web Server and OS disk 1 and OS disk 2 rolls.

   For each roll, add it to the frontend's roll repository:

   ```
   # rocks add roll <roll-name>.iso
   # rocks enable roll <roll-name>
   ```

2. Rebuild your distribution:

   ```
   # cd /export/rocks/install
   # rocks create distro
   # rocks create distro arch=i386
   ```

3. To prepare the frontend to cross-kickstart compute nodes via PXE, you first need to add the `rocks-boot` package from the i386 architecture:

   ```
   # rpm -i --force /export/rocks/install/rocks-dist/i386/RedHat/RPMS/rocks-boot-5.4.3-1.i386.rpm
   # cp /boot/kickstart/default/vmlinuz-5.4.3-i386 /tftpboot/pxelinux
   # cp /boot/kickstart/default/initrd.img-5.4.3-i386 /tftpboot/pxelinux
   ```

4. Then you need to add a new `bootaction`:

   ```
   # rocks add bootaction action="install i386" kernel="vmlinuz-5.4.3-i386" \
     ramdisk="initrd.img-5.4.3-i386" \
     args="ks ramdisk_size=150000 lang= devfs=nomount pxe kssendmac selinux=0 noipv6"
   ```

5. The next step requires you to PXE boot the cross-kickstarted node twice: the first is to discover the node, and a second time to install it.

   On the frontend, run insert-ethers to discover the compute node:

   ```
   # insert-ethers
   ```

   Then, PXE boot the compute node. After the compute node is discovered by insert-ethers, kill insert-ethers (by hitting the F9 key), then turn off the compute node.

Now instruct the node to use the "install i386" profile when it PXE boots (the command below assumes the compute node's name is compute-0-0):

```
# rocks set host installaction compute-0-0 action="install i386"
```

Now PXE boot the compute node again. The compute node will now install itself with the i386 distribution.

8.7. Adding Kernel Boot Parameters

There are two scenarios in which you need to add extra kernel boot parameters: 1) during installation, and 2) during normal boot. For both scenarios, we will use the example of adding the kernel boot parameter: "ucsd=rocks".

8.7.1. Adding Kernel Boot Parameters to the Installation Kernel

The boot *action* of a node is controlled by the Rocks command line. For example, the following lists what the nodes will do on their next PXE boot:

```
# rocks list host boot
HOST         ACTION
vi-2:        ------
compute-0-0: os
compute-0-1: os
```

We see that `compute-0-0` will boot the "os", that is, it will boot off its local disk. Another common action is "install" which means the node will boot the installation kernel and install itself on the node's next PXE boot.

To list all the possible boot actions for a node, execute:

```
# rocks list bootaction
ACTION             KERNEL                 RAMDISK             ARGS
install:           vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
install headless: vmlinuz-5.2-i386        initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
memtest:           kernel memtest         ------------------- ----------------------------------------
os:                localboot 0            ------------------- ----------------------------------------
pxeflash:          kernel memdisk bigraw pxeflash.img         keeppxe
rescue:            vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
```

To change a node's boot action to "install", execute:

```
# rocks set host boot compute-0-0 action="install"
```

Then we see that correct boot action has been applied:

```
# rocks list host boot
HOST         ACTION
vi-2:        -------
compute-0-0: install
compute-0-1: os
```

Now to change the kernel boot parameters for the installing kernel, we'll need to add an action. We'll create a new action called "install ucsd" and add our boot flag "ucsd=rocks" to the end of the kernel boot parameters:

```
# rocks add bootaction action="install ucsd" kernel="vmlinuz-5.2-i386" ramdisk="initrd.img-5.2-i386"
args="ks ramdisk_size=150000 lang= devfs=nomount pxe kssendmac selinux=0 noipv6 ucsd=rocks"
```

We now see that "install ucsd" is an available action:

```
# rocks list bootaction
ACTION            KERNEL                 RAMDISK             ARGS
install:          vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
install headless: vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
install ucsd:     vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
memtest:          kernel memtest         ------------------- ----------------------------------------
os:               localboot 0            ------------------- ----------------------------------------
pxeflash:         kernel memdisk bigraw pxeflash.img         keeppxe
rescue:           vmlinuz-5.2-i386       initrd.img-5.2-i386 ks ramdisk_size=150000 lang= devfs=nomou
```

To set the install action for compute-0-0 to "install ucsd", execute:

```
# rocks set host installaction compute-0-0 action="install ucsd"
```

Then we see that correct boot action has been applied:

```
# rocks list host
HOST          MEMBERSHIP CPUS RACK RANK RUNACTION INSTALLACTION
vi-2:         Frontend   1    0    0    os        install
compute-0-0: Compute    2    0    0    os        install ucsd
compute-0-1: Compute    2    0    1    os        install
```

The next time compute-0-0 is installed, it will use the "install ucsd" profile.

8.7.2. Adding Kernel Boot Parameters to the Running Kernel

To get the current parameters that will be appended to the default kernel boot parameters, execute:

```
# rocks report host bootflags
rocks-168: dom0_mem=1024M
compute-0-0: dom0_mem=1024M
```

Then, to add our boot flag of "ucsd=rocks", execute:

```
# rocks set host bootflags compute-0-0 flags="dom0_mem=1024M ucsd=rocks"
```

To check to make sure the flag has been added to the database, execute:

```
# rocks report host bootflags
rocks-168: dom0_mem=1024M
compute-0-0: dom0_mem=1024M ucsd=rocks
```

To apply the flags to the running kernel, you'll need to reinstall the compute nodes. For example, you'll need to reinstall compute-0-0 in order to apply "ucsd=rocks" to its kernel boot parameters. After compute-0-0 is reinstalled, you can check that the new kernel boot parameters have been applied by logging into compute-0-0 and executing:

```
# cat /proc/cmdline
```

```
ro root=LABEL=/ dom0_mem=1024M ucsd=rocks
```

Notes

1. http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.5/html/Deployment_Guide/s2-modules-bonding.html

Chapter 9. Command Reference

9.1. add

9.1.1. add appliance

rocks add appliance
{appliance} [graph=*string*] [membership=*string*] [node=*string*] [os=*string*] [public=*bool*]

Add an appliance specification to the database.

arguments

appliance

The appliance name (e.g., ’compute’, ’frontend’, ’nas’).

parameters

[graph=*string*]

The directory name of the graph XML files. The default is ’default’.

[membership=*string*]

The full membership name of the appliance. This name will be displayed in the appliances menu by insert-ethers (e.g., ’NAS Appliance’). If not supplied, the membership name is set to the appliance name.

[node=*string*]

The name of the root XML node (e.g., ’compute’, ’nas’, ’viz-tile’). If not supplied, the node name is set to the appliance name.

[os=*string*]

The OS that the appliance type can support. Some appliances can support both linux and sunos, where as others can support only one of the two. Acceptable values are ’linux’ or ’sunos’. Defaults to ’linux’

[public=*bool*]

True means this appliance will be displayed by ’insert-ethers’ in the Appliance menu. The default is ’yes’.

examples

rocks add appliance nas membership="NAS Appliance" node=nas graph=default public=yes

rocks add appliance tile membership=Tile node=viz-tile graph=default public=yes

9.1.2. add appliance attr

rocks add appliance attr {appliance} {attr} {value} [attr=*string*] [value=*string*]

Adds an attribute to an appliance and sets the associated values

arguments

appliance

Name of appliance

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks add appliance attr compute sge False

Sets the sge attribution to False for compute appliances

rocks add appliance attr compute sge attr=cpus value=2

same as above

related commands

list appliance attr

list host attr

remove appliance attr

remove host attr

set host attr

9.1.3. add appliance route

rocks add appliance route {os} {address} {gateway} [netmask=*string*]

Add a route for an appliance type in the cluster

arguments

os

The OS type (e.g., 'linux', 'sunos', etc.). This argument is required.

address

Host or network address

gateway

Network or device gateway

parameters

[netmask=*string*]

Specifies the netmask for a network route. For a host route this is not required and assumed to be 255.255.255.255

9.1.4. add attr

rocks add attr {attr} {value} [attr=*string*] [value=*string*]

Adds a global attribute for all nodes

arguments

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks add attr sge False

Adds the sge attribution and sets it to False.

related commands

list attr

remove attr

9.1.5. add bootaction

rocks add bootaction [action=*string*] [args=*string*] [kernel=*string*] [ramdisk=*string*]

Add a bootaction specification to the system.

parameters

[action=*string*]

Label name for the bootaction. You can see the bootaction label names by executing: 'rocks list bootaction [host(s)]'.

[args=*string*]

The second line for a pxelinux definition (e.g., ks ramdisk_size=150000 lang= devfs=nomount pxe kssendmac selinux=0)

[kernel=*string*]

The name of the kernel that is associated with this boot action.

[ramdisk=*string*]

The name of the ramdisk that is associated with this boot action.

examples

rocks add bootaction action=os kernel="localboot 0"

Add the 'os' bootaction.

rocks add bootaction action=memtest command="memtest"

Add the 'memtest' bootaction.

9.1.6. add distribution

rocks add distribution {distribution}

Add a distribution specification to the database.

arguments

distribution

Name of the new distribution.

examples

rocks add distribution rocks-dist

Adds the distribution named "rocks-dist" into the database.

9.1.7. add firewall

rocks add firewall {category=index} [action=*string*] [chain=*string*] [network=*string*] [output-network=*string*] [protocol=*string*] [rulename=*string*] [rulesrc=*string*] [service=*string*]

Add a firewall rule to the a category in cluster.

arguments

category=index

[global,os,appliance,host]=index. Must precede all other a=b parameters Apply rule to index (member) of category. e.g. os=linux, appliance=login, or host=compute-0-0. global, global=, and global=global all refer to the global category

parameters

[action=*string*]

The iptables 'action' this rule should be applied to (e.g., ACCEPT, REJECT, DROP).

[chain=*string*]

The iptables 'chain' this rule should be applied to (e.g., INPUT, OUTPUT, FORWARD).

[network=*string*]

The network this rule should be applied to. This is a named network (e.g., 'private') and must be one listed by the command 'rocks list network'. To have this firewall rule apply to all networks, specify the keyword 'all'.

[output-network=*string*]

The output network this rule should be applied to. This is a named network (e.g., 'private') and must be one listed by the command 'rocks list network'.

[protocol=*string*]

The protocol associated with the rule. For example, "tcp" or "udp". To have this firewall rule apply to all protocols, specify the keyword 'all'.

[rulename=*string*]

User-defined name of rule. Required. Firewall rules are ordered lexicographically.

[rulesrc=*string*]

system or custom. Default is 'custom'. Rules defined as 'system' are not dumped with rocks dump firewall.

[service=*string*]

The service identifier, port number or port range. For example "www", 8080 or 0:1024. To have this firewall rule apply to all services, specify the keyword 'all'.

examples

rocks add firewall appliance=login rulename=ACCEPT-SSH network=public service="ssh" protocol="tcp" action="ACCEPT" chain="INPUT" flags="-m state --state NEW"

Accept TCP packets for the ssh service on the public network on the INPUT chain and apply the "-m state --state NEW" flags to the rule. Apply the rule to login appliances (appliance=login) Name the rule ACCEPT-SSH If 'eth1' is associated with the public network, this will be translated as the following iptables rule: "-A INPUT -i eth1 -p tcp --dport ssh -m state --state NEW -j ACCEPT"

rocks add firewall global rulename=ACCEPT-PRIVATE network=private service="all" protocol="all" action="ACCEPT" chain="INPUT"

Accept all protocols and all services on the private network on the INPUT chain. Apply this rule to all nodes in the cluster (global) If 'eth0' is the private network, then this will be translated as the following iptables rule: "-A INPUT -i eth0 -j ACCEPT"

rocks add firewall host=compute-0-0 rulename=ZZDRACONIAN network="all" service="all" protocol="all" action="DROP" chain="INPUT"

DROP all non-matched packets Apply this rule to host compute-0-0 (host=compute-0-0) rule will be named ZZDRACONIAN This will drop all non-matched packets that have not been previously accepted Known as a draconian firewall rule.

9.1.8. add host

rocks add host {host} [cpus=*int*] [membership=*string*] [os=*string*] [rack=*int*] [rank=*int*]

Add an new host to the cluster.

arguments

host

A single host name. If the hostname is of the standard form of basename-rack-rank the default values for the membership, rack, and rank parameters are taken from the hostname.

parameters

[cpus=*int*]

Number of CPUs (cores) in the given host. If not provided the default of 1 CPU is inserted into the database.

[membership=*string*]

Appliance membership name. If not provided and the host name is of the standard form the membership is taken from the basename of the host.

[os=*string*]

The operating system name. The default is: linux.

[rack=*int*]

The number of the rack where the machine is located. The convention in Rocks is to start numbering at 0. If not provided and the host name is of the standard form the rack number is taken from the host name.

[rank=*int*]

The position of the machine in the rack. The convention in Rocks is to number from the bottom of the rack to the top starting at 0. If not provided and the host name is of the standard form the rank number is taken from the host name.

examples

rocks add host compute-0-1

Adds the host "compute-0-0" to the database with 1 CPU, a membership name of "compute", a rack number of 0, and rank of 1.

\# rocks add host frontend rack=0 rank=0 membership=Frontend

Adds the host "frontend" to the database with 1 CPU, a membership name of "Frontend", a rack number of 0, and rank of 1.

related commands

add host interface

9.1.9. add host alias

rocks add host alias {host} {name} [name=*string*]

Adds an alias to a host

arguments

host

Host name of machine

name

The alias name for the host.

parameters

[name=*string*]

Can be used in place of the name argument.

examples

\# rocks add host alias compute-0-0 c-0-0

Adds the alias 'c-0-0' to the host 'compute-0-0'.

\# rocks add host alias compute-0-0 name=c-0-0

Same as above.

9.1.10. add host attr

rocks add host attr {host} {attr} {value} [attr=*string*] [value=*string*]

Adds an attribute to a host and sets the associated values

arguments

host

Host name of machine

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks add host attr compute-0-0 cpus 2

Sets the number of cpus of compute-0-0 to 2

rocks add host attr compute-0-0 attr=cpus value=2

same as above

related commands

list host attr

remove host attr

9.1.11. add host bonded

rocks add host bonded
{host} [channel=*string*] [interfaces=*string*] [ip=*string*] [name=*string*] [network=*string*]

Add a channel bonded interface for a host

arguments

host

Host name of machine

parameters

[channel=*string*]

The channel name (e.g., "bond0").

[interfaces=*string*]

The physical interfaces that will be bonded. The interfaces can be a comma-separated list (e.g., "eth0,eth1") or a space-separated list (e.g., "eth0 eth1").

[ip=*string*]

The IP address to assign to the bonded interface.

[name=*string*]

The host name associated with the bonded interface. If name is not specified, then the interface get the internal host name (e.g., compute-0-0).

[network=*string*]

The network to be assigned to this interface. This is a named network (e.g., 'private') and must be listable by the command 'rocks list network'.

examples

rocks add host bonded compute-0-0 channel=bond0 interfaces=eth0,eth1 ip=10.1.255.254 network=private

Adds a bonded interface named "bond0" to compute-0-0 by bonding the physical interfaces eth0 and eth1, it assigns the IP address 10.1.255.254 to bond0 and it associates this interface to the private network.

9.1.12. add host interface

rocks add host interface {host} {iface} [iface=*string*] [ip=*string*] [mac=*string*] [module=*string*] [name=*string*] [subnet=*string*] [vlan=*string*]

Adds an interface to a host and sets the associated values

arguments

host

Host name of machine

iface

The interface name on the host (e.g., ’eth0’, ’eth1’)

parameters

[iface=*string*]

Can be used in place of the iface argument.

[ip=*string*]

The IP address to assign to the interface (e.g., ’192.168.1.254’)

[mac=*string*]

The MAC address of the interface (e.g., ’00:11:22:33:44:55’)

[module=*string*]

The device driver name (or module) of the interface (e.g., ’e1000’)

[name=*string*]

The name to assign to the interface

[subnet=*string*]

The name of the subnet to assign to the interface (e.g., ’private’)

[vlan=*string*]

The VLAN ID to assign the interface

examples

rocks add host interface compute-0-0 eth1 ip=192.168.1.2 subnet=private name=fast-0-0

rocks add host interface compute-0-0 iface=eth1 ip=192.168.1.2 subnet=private name=fast-0-0

same as above

related commands

set host interface iface

set host interface ip

set host interface mac

set host interface module

set host interface name

set host interface subnet

9.1.13. add host key

rocks add host key {host} [key=*string*]

Add a public key for a host. One use of this public key is to authenticate messages sent from remote services.

arguments

host

Host name of machine

parameters

[key=*string*]

A public key. This can be the actual key or it can be a path name to a file that contains a public key (e.g., /tmp/public.key).

9.1.14. add host route

rocks add host route {host} {address} {gateway} [netmask=*string*]

Add a route for a host

arguments

host

Host name of machine

address

Host or network address

gateway

Network or device gateway

parameters

[netmask=*string*]

Specifies the netmask for a network route. For a host route this is not required and assumed to be 255.255.255.255

9.1.15. add host sec_attr

rocks add host sec_attr {host} [attr=*string*] [crypted=*boolean*] [enc=*string*] [value=*string*]

Add a secure attribute to the database. The user also has the option of not supplying the value on the command line. The command will prompt the user to enter the secure attribute value, and will not echo this value on screen. If the user chooses to enter the value for the secure attribute by prompt, then the value entered must be in an unencrypted form.

arguments

host

Host name of machine

parameters

[attr=*string*]

Name of the attribute

[crypted=*boolean*]

Is "value" already crypted or not

[enc=*string*]

Encryption scheme to use to crypt the value. Currently supported values are "sha", "crypt", "portable".

[value=*string*]

Value of the attribute

examples

rocks add host sec_attr compute-0-0 attr=db_pw value=DatabasePassword crypted=false enc=sha

Sets a secure attribute called db_pw to the crypted value of "DatabasePassword" using the sha1 encoding schema.

rocks add host sec_attr compute-0-0 attr=db_pw value=77e6674e6d71f898d5fc79424117c86731ca7498 crypted=true

Same as above

9.1.16. add network

rocks add network {name} {subnet} {netmask} [dnszone=*string*] [mtu=*string*] [netmask=*string*] [servedns=*boolean*] [subnet=*string*]

Add a network to the database. By default both the "public" and "private" networks are already defined by Rocks.

arguments

name

Name of the new network.

subnet

The IP network address for the new network.

netmask

The IP network mask for the new network.

parameters

[dnszone=*string*]

The Domain name or the DNS Zone name to use for all hosts of this particular subnet. Default is set to the name of the subnet

[mtu=*string*]

The MTU for the new network. Default is 1500.

[netmask=*string*]

Can be used in place of the netmask argument.

[servedns=*boolean*]

Parameter to decide whether this zone will be served by the nameserver on the frontend.

[subnet=*string*]

Can be used in place of the subnet argument.

examples

rocks add network optiputer 192.168.1.0 255.255.255.0

Adds the optiputer network address of 192.168.1.0/255.255.255.0.

rocks add network optiputer subnet=192.168.1.0 netmask=255.255.255.0 mtu=9000 dnszone="optiputer.net" servedns=true

Same as above, but set the MTU to 9000.

9.1.17. add os attr

rocks add os attr {os} {attr} {value} [attr=*string*] [value=*string*]

Adds an attribute to an os and sets the associated values

arguments

os

Name of os

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks add os attr linux sge False

Sets the sge attribution to False for linux nodes

9.1.18. add os route

rocks add os route {os} {address} {gateway} [netmask=*string*]

Add a route for an OS type

arguments

os

The OS type (e.g., 'linux', 'sunos', etc.). This argument is required.

address

Host or network address

gateway

Network or device gateway

parameters

[netmask=`string`]

Specifies the netmask for a network route. For a host route this is not required and assumed to be 255.255.255.255

9.1.19. add roll

rocks add roll [roll...] [clean=`bool`]

Add Roll ISO images to this machine's roll directory. This command copies all files in the ISOs to a directory under /export/rocks/install/rolls.

arguments

[roll]

A list of Roll ISO images to add to /export/rocks/install/rolls. If no list is supplied, then if a roll is mounted on /mnt/cdrom, it will be copied into /export/rocks/install/rolls.

parameters

[clean=`bool`]

If set, then remove all files from any existing rolls of the same name, version, and architecture before copying the contents of the Rolls onto the local disk. This parameter should not be set when adding multi-CD Rolls such as the OS Roll, but should be set when adding single Roll CDs such as the Grid Roll.

examples

rocks add roll clean=1 kernel*iso

Adds the Kernel Roll to local Roll directory. Before the Roll is added the old Kernel Roll packages are removed from the Roll directory.

rocks add roll kernel*iso pvfs2*iso ganglia*iso

Added the Kernel, PVFS, and Ganglia Rolls to the local Roll directory.

related commands

create roll

disable roll

enable roll

list roll

remove roll

9.1.20. add route

rocks add route {address} {gateway} [netmask=*string*]

Add a route for all hosts in the cluster

arguments

address

Host or network address

gateway

Network (e.g., IP address), subnet name (e.g., 'private', 'public'), or a device gateway (e.g., 'eth0).

parameters

[netmask=*string*]

Specifies the netmask for a network route. For a host route this is not required and assumed to be 255.255.255.255

9.1.21. add sec_attr

rocks add sec_attr {attr} [crypted=*boolean*] [enc=*string*] [value=*string*]

Add a secure attribute to the database. The user also has the option of not supplying the value on the command line. The command will prompt the user to enter the secure attribute value, and will not echo this value on screen. If the user chooses to enter the value for the secure attribute by prompt, then the value entered must be in an unencrypted form.

arguments

attr

Name of the attribute

parameters

[crypted=*boolean*]

Is "value" already crypted or not

[enc=*string*]

Encryption scheme to use to crypt the value. Currently supported values are "sha", "crypt", "portable".

[value=*string*]

Value of the attribute

examples

rocks add sec_attr db_pw value=DatabasePassword crypted=false enc=sha

Sets a secure attribute called db_pw to the crypted value of "DatabasePassword" using the sha1 encoding schema.

rocks add sec_attr db_pw value=77e6674e6d71f898d5fc79424117c86731ca7498 crypted=true

Same as above

9.1.22. add var

rocks add var
{service} {component} {value} [appliance=*string*] [component=*string*] [service=*string*] [value=*string*]

Add variables to the Rocks key/value database. Variables are defined as quad of (Appliance, Service, Component, Value). Within a node XML file, values can be retrieved as <var name="Service_Component"/>.

arguments

service

Defines the service name. e.g., service=Kickstart.

component

Defines the component name. e.g. component=PublicDNS.

value

Defines the value for the variable.

parameters

[appliance=*string*]

If supplied, restricts to the named appliance. See 'rocks list appliance' for a listing of appliances.

[component=*string*]

Can be used in place of component argument.

[service=*string*]

Can be used in place of service argument.

[value=*string*]

Can be used in place of value argument.

examples

rocks add var service=Condor component=Master value=localhost

Add the variable name <var name="Condor_Master"/> to 'localhost'.

rocks add var Condor Master localhost

Same as above.

rocks add var service=Condor component=Master value=localhost appliance=compute

Add the variable name <var name="Condor_Master"/> to 'localhost' and associate it with only compute appliances.

9.2. config

9.2.1. config host interface

rocks config host interface {host} [flag=*string*] [iface=*string*] [mac=*string*] [module=*string*]

Adds host interfaces to the database. This command should only be called from a post section in a kickstart file.

arguments

host

Host name of machine

parameters

[flag=*string*]

Flags for the interfaces. If flags for multiple interfaces are supplied, then they must be comma-separated.

[iface=*string*]

Interface names (e.g., "eth0"). If multiple interfaces are supplied, then they must be comma-separated.

[mac=*string*]

MAC addresses for the interfaces. If multiple MACs are supplied, then they must be comma-separated.

[module=*string*]

Driver modules to be loaded for the interfaces. If multiple modules are supplied, then they must be comma-separated.

9.3. create

9.3.1. create distro

rocks create distro [arch=*string*] [dist=*string*] [rolls=*string*] [root=*string*] [version=*string*]

Create a Rocks distribution. This distribution is used to install Rocks nodes.

parameters

[arch=*string*]

The architecture of the distribution. The default is the native architecture of the machine.

[dist=*string*]

The directory name of the distribution. The default is: "rocks-dist".

[rolls=*string*]

A list of rolls that should be included in the distribution. This must be a list separated by spaces of the form: rollname,version. For example: rolls="CentOS,5.0 kernel,5.0". The default is to include all the enabled rolls for the native architecture. To get a list of enabled rolls, execute: "rocks list roll".

[root=*string*]

The path prefix location of the rolls. The default is: /export/rocks/install.

[version=*string*]

The version of the distribution. The default is the native version of the machine.

examples

rocks create distro

Create a distribution in the current directory.

9.3.2. create keys

rocks create keys [key=*string*] [passphrase=*boolean*]

Create a RSA private/public key pair. These keys can be used to control the power for host and to open a console to VM. The private key will be stored in the specified by the ’key’ parameter and the public key will be written to standard out.

parameters

[key=*string*]

The filename that will be used to store the private key.

[passphrase=*boolean*]

Set this to ’no’ if you want a passphraseless private key. The default is ’yes’.

9.3.3. create mirror

rocks create mirror {path} [arch=*string*] [rollname=*string*] [version=*string*]

Create a Roll ISO image from the packages found in the repository located at ’URL’.

arguments

path

The network location of the repository of packages.

parameters

[arch=*string*]

Architecture of the mirror. (default = the architecture of of the OS running on this machine).

[rollname=*string*]

The base name for the created Roll. (default = ’updates’).

[version=*string*]

The version number of the created Roll. (default = the version of Rocks running on this machine).

examples

rocks create mirror http://mirrors.kernel.org/centos/4.5/updates/i386/RPMS rollname=updates version=4.5

Will mirror all the packages found under the URL http://mirrors.kernel.org/centos/4.5/updates/i386/RPMS and will create a Roll ISO image named 'updates-4.5-0.i386.disk1.iso'.

9.3.4. create new roll

rocks create new roll {version} {name} {color} [color=*string*] [name=*string*] [version=*string*]

Create a new roll from a template.

arguments

version

name

color

parameters

[color=*string*]

[name=*string*]

[version=*string*]

examples

rocks create new roll

9.3.5. create package

rocks create package {directory} [prefix=*string*] [release=*string*] [version=*string*]

Create a RedHat or Solaris package from a given directory. The package will install files in either the same location as the given directory, or a combination of the directory basename and the provided prefix.

arguments

directory

The source directory of the files used to create the OS-specific package.

parameters

[prefix=*string*]

The prefix pathname prepended to the base name of the source directory.

[release=*string*]

Release number of the created package (default is '1')

[version=*string*]

Version number of the created package (default is '1.0')

examples

rocks create package /opt/stream stream

Create a package named stream in the current directory using the contents of the /opt/stream directory. The resulting package will install its files at /opt/stream.

rocks create package /opt/stream localstream prefix=/usr/local

Create a package named localstream in the current directory using the contents of the /opt/stream directory. The resulting package will install its files at /usr/local/stream.

9.3.6. create roll

rocks create roll {roll...}

Create a roll. You may specify either a single XML file to build one Roll or a list of ISO files to build a Meta Roll.

arguments

roll

Either a list of Roll ISO files or the name of a single Roll XML description file. If a list of Roll ISO files to be merge together into a single Roll. Otherwise the single argument is assumed to be the name of the XML file generated by the top level Makefile in the Roll's source.

examples

rocks create roll roll-base.xml

Creates the Rocks Base Roll from the roll-base.xml description file.

rocks create roll base*iso kernel*iso

Create a composite Roll from a list of Roll ISOs.

related commands

add roll

disable roll

enable roll

list roll

remove roll

9.4. disable

9.4.1. disable roll

rocks disable roll {roll...} [arch=*string*] [version=*string*]

Disable an available roll. The roll must already be copied on the system using the command "rocks add roll".

arguments

roll

List of rolls to disable. This should be the roll base name (e.g., base, hpc, kernel).

parameters

[arch=*string*]

The architecture to disable this roll for. If no architecture is supplied, then the roll will be disabled for all architectures.

[version=*string*]

The version number of the roll to be disabled. If no version number is supplied, then all versions of a roll will be disabled.

examples

rocks disable roll kernel

Disable the kernel roll

rocks disable roll ganglia version=5.0 arch=i386

Disable version 5.0 the Ganglia roll for i386 nodes

related commands

add roll

create roll

enable roll

list roll

remove roll

9.5. dump

9.5.1. dump

rocks dump

The top level dump command is used to recursively call all the dump commands in the correct order. This is used to create the restore roll.

examples

$ rocks dump

Recursively call all dump commands.

9.5.2. dump appliance

rocks dump appliance [appliance...]

Outputs info (as rocks commands) about the appliances defined in the cluster database.

arguments

[appliance]

Optional list of appliance names. If no appliance names are supplied, then info about all appliances is output.

examples

$ rocks dump appliance

Dump all known appliances.

9.5.3. dump appliance attr

rocks dump appliance attr [appliance]

Dump the set of attributes for appliances.

arguments

[appliance]

Name of appliance

examples

$ rocks dump appliance attr compute

List the attributes for compute appliances

9.5.4. dump appliance route

rocks dump appliance route

9.5.5. dump attr

rocks dump attr

9.5.6. dump firewall

rocks dump firewall

9.5.7. dump host

rocks dump host [host...]

Dump the host information as rocks commands.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

examples

$ rocks dump host compute-0-0

Dump host compute-0-0 information.

$ rocks dump host compute-0-0 compute-0-1

Dump host compute-0-0 and compute-0-1 information.

$ rocks dump host

Dump all hosts.

9.5.8. dump host attr

rocks dump host attr [host]

Dump the set of attributes for hosts.

arguments

[host]

Host name of machine

examples

$ rocks dump host attr compute-0-0

Dump the attributes for compute-0-0.

9.5.9. dump host boot

rocks dump host boot

9.5.10. dump host interface

rocks dump host interface [host...]

Dump the host interface information as rocks commands.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

examples

$ rocks dump host interface compute-0-0

Dump the interfaces for compute-0-0.

$ rocks dump host interface compute-0-0 compute-0-1

Dump the interfaces for compute-0-0 and compute-0-1.

$ rocks dump host interface

Dump all interfaces.

related commands

add host interface

9.5.11. dump host key

rocks dump host key

9.5.12. dump host roll

rocks dump host roll

9.5.13. dump host route

rocks dump host route

9.5.14. dump network

rocks dump network [network...]

Dump the network information as rocks commands.

arguments

[network]

Zero, one or more network names. If no network names are supplied, information for all networks will be listed.

examples

$ rocks dump network

Dump all network info.

$ rocks dump network public

Dump network info the 'public' network.

9.5.15. dump os attr

rocks dump os attr

9.5.16. dump os route

rocks dump os route

9.5.17. dump route

rocks dump route

9.6. enable

9.6.1. enable roll

rocks enable roll {roll...} [arch=*string*] [version=*string*]

Enable an available roll. The roll must already be copied on the system using the command "rocks add roll".

arguments

roll

List of rolls to enable. This should be the roll base name (e.g., base, hpc, kernel).

parameters

[arch=*string*]

The architecture to enable this roll for. If no architecture is supplied, then the roll will be enabled for all architectures.

[version=*string*]

The version number of the roll to be enabled. If no version number is supplied, then all versions of a roll will be enabled.

examples

rocks enable roll kernel

Enable the kernel roll

rocks enable roll ganglia version=5.0 arch=i386

Enable version 5.0 the Ganglia roll for i386 nodes

related commands

add roll

create roll

disable roll

list roll

remove roll

9.7. help

9.7.1. help

rocks help {command}

List help for the command line client. With no arguments it lists all the commands available. Otherwise it will list the subset of command with the specified string (see examples).

arguments

command

The substring matched against all commands.

examples

$ rocks help

Alias for ’rocks list help’

$ rocks help viz

Lists all the commands with the string ’viz’ in the name.

$ rocks help list host

Lists all the commands with the string ’list host’ in the name.

9.8. iterate

9.8.1. iterate host

rocks iterate host [host...] [command] [command=*string*]

Iterate sequentially over a list of hosts. This is used to run a shell command on the frontend with with ’%’ wildcard expansion for every host specified.

arguments

[host]

Zero, one or more host names. If no host names are supplied iterate over all hosts except the frontend.

[command]

The shell command to be run for each host. The ’%’ character is used as a wildcard to indicate the hostname. Quoting of the ’%’ to expand to a literal is accomplished with ’%%’.

parameters

[command=*string*]

Can be used in place of the command argument.

examples

$ rocks iterate host compute "scp file %:/tmp/"

Copies file to the /tmp directory of every compute node

$ rocks iterate host compute command="scp file %:/tmp/"

Same as above

9.9. list

9.9.1. list appliance

rocks list appliance [appliance...]

Lists the appliances defined in the cluster database.

arguments

[appliance]

Optional list of appliance names.

examples

$ rocks list appliance

List all known appliances.

9.9.2. list appliance attr

rocks list appliance attr [appliance]

Lists the set of attributes for appliances.

arguments

[appliance]

Name of appliance

examples

$ rocks list appliance attr compute

List the attributes for compute appliances

9.9.3. list appliance route

rocks list appliance route

9.9.4. list appliance xml

rocks list appliance xml [appliance...]

Lists the XML profile for a given appliance type. This is useful for high level debugging but will be missing any host specific variables. It cannot be used to pass into 'rocks list host profile' to create a complete Kickstart/Jumpstart profile.

arguments

[appliance]

Optional list of appliance names.

examples

$ rocks list appliance xml compute

Lists the XML profile for a compute appliance.

$ rocks list appliance xml

Lists the XML profile for all appliance types.

9.9.5. list attr

rocks list attr

Lists the set of global attributes.

examples

$ rocks list attr

List the global attributes.

9.9.6. list bootaction

rocks list bootaction

Lists the set of boot actions for hosts. Each boot action is a label that points to a command string. The command string is placed into a host-specific pxelinux configuration file. Example labels are 'install' and 'os' which point to command strings used to install and boot hosts respectively.

examples

$ rocks list bootaction

List the boot actions available for all known hosts.

9.9.7. list distribution

rocks list distribution [distribution...]

Lists the distributions defined in the cluster database.

arguments

[distribution]

Optional list of distribution names.

examples

$ rocks list distribution

List all known distribution definitions.

9.9.8. list firewall

rocks list firewall [category=index] [maxwidth=*integer*]

List the firewall rules for a particular category

arguments

[category=index]

[global,os,appliance,host]=index. list rules index (member) of category. e.g. os=linux, appliance=login, or host=compute-0-0. global, global=, and global=global all refer to the global category.

parameters

[maxwidth=*integer*]

Maximum width of comment and flags field. Default is 24.

9.9.9. list help

rocks list help [subdir=*string*]

The Help Command print the usage of all the registered Commands.

parameters

[subdir=*string*]

Relative of Python commands for listing help. This is used internally only.

examples

$ rocks list help

List help for all commands

$ rocks list help subdir=list/host

List help for all commands under list/host

9.9.10. list host

rocks list host [host...]

List the membership, CPU count and physical position info for a list of hosts.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host compute-0-0

List info for compute-0-0.

$ rocks list host

List info for all known hosts.

9.9.11. list host alias

rocks list host alias [host...]

Lists the aliases for a host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, aliases for all the known hosts is listed.

examples

$ rocks list host alias compute-0-0

List the aliases for compute-0-0.

$ rocks list host membership

List the aliases for all known hosts.

9.9.12. list host appliance

rocks list host appliance [host...]

Lists the appliance assignments for hosts. For each host supplied on the command line, this command prints the hostname and appliance assignment for that host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

examples

$ rocks list host appliance compute-0-0

List the appliance assignment for compute-0-0.

$ rocks list host appliance compute-0-0 compute-0-1

List the appliance assignments for compute-0-0 and compute-0-1.

9.9.13. list host attr

rocks list host attr [host]

Lists the set of attributes for hosts.

arguments

[host]

Host name of machine

examples

$ rocks list host attr compute-0-0

List the attributes for compute-0-0.

9.9.14. list host boot

rocks list host boot [host...]

Lists the current bot action for hosts. For each host supplied on the command line, this command prints the hostname and boot action for that host. The boot action describes what the host will do the next time it is booted.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host boot compute-0-0

List the current boot action for compute-0-0.

$ rocks list host boot

List the current boot action for all known hosts.

9.9.15. list host firewall

rocks list host firewall [host...] [maxwidth=`integer`]

List the current firewall rules for the named hosts.

arguments

[host]

Zero, one or more host names. If no host names are supplied, the firewall rules for all the known hosts are listed.

parameters

[maxwidth=`integer`]

Maximum width of comment and flags field. Default is 24.

9.9.16. list host graph

rocks list host graph [host...] [arch=*string*] [basedir=*string*]

For each host, output a graphviz script to produce a diagram of the XML configuration graph. If no hosts are specified, a graph for every known host is listed.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

parameters

[arch=*string*]

Optional. If specified, generate a graph for the specified CPU type. If not specified, then 'arch' defaults to this host's architecture.

[basedir=*string*]

Optional. If specified, the location of the XML node files.

examples

$ rocks list host graph compute-0-0

Generates a graph for compute-0-0

9.9.17. list host installfile

rocks list host installfile [section=*string*]

Process an XML-based installation file and output an OS-specific installation file (e.g., a kickstart or jumpstart file).

parameters

[section=*string*]

Which section within the XML installation file to process (e.g., "kickstart", "begin", etc.).

examples

rocks list host installfile section="kickstart"

Output a RedHat-compliant kickstart file.

9.9.18. list host interface

rocks list host interface [host...]

Lists the interface definitions for hosts. For each host supplied on the command line, this command prints the hostname and interface definitions for that host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host interface compute-0-0

List network interface info for compute-0-0.

$ rocks list host interface

List network interface info for all known hosts.

9.9.19. list host key

rocks list host key [host...]

List the public keys for hosts.

arguments

[host]

Zero, one or more host names. If no host names are supplied, information for all hosts will be listed.

9.9.20. list host macs

rocks list host macs {host} [key=*string*] [status=*bool*]

Get a list of MAC addresses for all the hosts that are associated with a virtual cluster.

arguments

host

Host name of machine. This host name should be the name of the virtual frontend for the virtual cluster that you want the MAC addresses for.

parameters

[key=*string*]

A private key that will be used to authenticate the request. This should be a file name that contains the private key.

[status=*bool*]

If true, then for each VM-based cluster node, output the VM's status (e.g., 'active', 'paused', etc.).

9.9.21. list host membership

rocks list host membership [host...]

Lists the membership assignments for hosts. For each host supplied on the command line, this command prints the hostname and membership assignment for that host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host membership compute-0-0

List the membership assignment for compute-0-0.

$ rocks list host membership

List the membership assignment for all known hosts.

9.9.22. list host partition

rocks list host partition [host...]

Lists the partitions for hosts. For each host supplied on the command line, this command prints the hostname and partitions for that host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host partition compute-0-0

List partition info for compute-0-0.

$ rocks list host partition

List partition info for known hosts.

9.9.23. list host profile

rocks list host profile [host...]

Outputs a XML wrapped Kickstart/Jumpstart profile for the given hosts. If not, profiles are listed for all hosts in the cluster. If input is fed from STDIN via a pipe, the argument list is ignored and XML is read from STDIN. This command is used for debugging the Rocks configuration graph.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

rocks list host profile compute-0-0

Generates a Kickstart/Jumpstart profile for compute-0-0.

rocks list host xml compute-0-0 | rocks list host profile

Does the same thing as above but reads XML from STDIN.

9.9.24. list host roll

rocks list host roll [host]

List the assignment of rolls to frontend appliances.

arguments

[host]

Host name of machine

examples

$ rocks list host roll frontend-0-0-0

List the rolls assigned to frontend-0-0-0.

9.9.25. list host route

rocks list host route [host]

List the static routes that are assigned to a host.

arguments

[host]

Host name of machine

examples

$ rocks list host route compute-0-0

List the static routes assigned to compute-0-0.

9.9.26. list host sec_attr

rocks list host sec_attr {host}

Lists the secure attributes for a given host

arguments

host

Hostname(s)

9.9.27. list host xml

rocks list host xml [host...]

Lists the monolithic XML configuration file for hosts. For each host supplied on the command line, this command prints the hostname and XML file configuration for that host. This is the same XML configuration file that is sent back to a host when a host begins it’s installation procedure.

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks list host xml compute-0-0

List the XML configuration file for compute-0-0.

$ rocks list host xml

List the XML configuration files for all known hosts.

9.9.28. list license

rocks list license

List the Rocks copyright.

examples

$ rocks list license

List the Rocks copyright.

9.9.29. list membership

rocks list membership [membership...]

Lists the memberships defined in the cluster database.

arguments

[membership]

Optional. A list of membership names. If no membership names are supplied, all the known memberships are listed.

examples

$ rocks list membership

List all known membership definitions.

9.9.30. list network

rocks list network [network...]

List the defined networks for this system.

arguments

[network]

Zero, one or more network names. If no network names are supplied, info about all the known networks is listed.

examples

$ rocks list network private

List network info for the network named 'private'.

$ rocks list network

List info for all defined networks.

9.9.31. list node xml

rocks list node xml
[attrs=*string*] [basedir=*string*] [eval=*bool*] [gen=*string*] [missing-check=*bool*] [roll=*string*]

Lists the XML configuration information for a host. The graph traversal for the XML output is rooted at the XML node file specified by the 'node' argument. This command executes the first pre-processor pass on the configuration graph, performs all variable substitutions, and runs all eval sections.

parameters

[attrs=*string*]

A list of attributes. This list must be in python dictionary form, e.g., attrs="{ 'os': 'linux', 'arch' : 'x86_64' }"

[basedir=*string*]

If specified, the location of the XML node files.

[eval=*bool*]

If set to 'no', then don't execute eval sections. If not supplied, then execute all eval sections.

[gen=*string*]

If set, the use the supplied argument as the program for the 2nd pass generator. If not supplied, then use 'kgen'.

[missing-check=*bool*]

If set to 'no', then disable errors regarding missing nodes. If not supplied, then print messages about missing nodes.

[roll=*string*]

If set, only expand nodes from the named roll. If not supplied, then the all rolls are used.

examples

$ rocks list node xml compute

Generate the XML graph starting at the XML node named 'compute.xml'.

9.9.32. list os attr

rocks list os attr [os]

Lists the set of attributes for OSes.

arguments

[os]

Name of OS (e.g. "linux", "sunos")

examples

$ rocks list os attr linux

List the attributes for the Linux OS

9.9.33. list os route

rocks list os route

9.9.34. list roll

rocks list roll [roll...]

List the status of available rolls.

arguments

[roll]

List of rolls. This should be the roll base name (e.g., base, hpc, kernel). If no rolls are listed, then status for all the rolls are listed.

examples

$ rocks list roll kernel

List the status of the kernel roll

$ rocks list roll

List the status of all the available rolls

related commands

add roll

create roll

disable roll

enable roll

remove roll

9.9.35. list roll command

rocks list roll command [roll...]

List the commands provided by a roll.

arguments

[roll]

List of rolls. This should be the roll base names (e.g., base, hpc, kernel). If no rolls are listed, then commands for all the rolls are listed.

examples

$ rocks list roll command base

Returns the the list of commands installed by the Rocks base Roll.

9.9.36. list route

rocks list route

9.9.37. list var

rocks list var [appliance=*string*] [component=*string*] [service=*string*]

Lists variables in the Rocks key/value database. Variables are defined as a quad of (Appliance, Service, Component, Value). Within a node XML file, values can be retrieved as <var name="Service_Component"/>.

parameters

[appliance=*string*]

If supplied, restricts listing to this appliance. Using appliance=global will list only global values.

[component=*string*]

If supplied, restricts listing to this component.

[service=*string*]

If supplied, restricts listing to this service.

examples

$ rocks list var

List all Rocks variables.

$ rocks list var service=Kickstart

List all Rocks variables associated with the 'Kickstart' service.

$ rocks list var service=Info component=RocksVersion

List the Rocks variable for Info_RocksVersion.

9.10. remove

9.10.1. remove appliance

rocks remove appliance {name}

Remove an appliance definition from the system. This can be called with just the appliance or it can be further qualified by supplying the root XML node name and/or the graph XML file name.

arguments

name

The name of the appliance.

examples

rocks remove appliance compute

Removes the compute appliance from the database.

9.10.2. remove appliance attr

rocks remove appliance attr {appliance} {attr} [attr=*string*]

Remove an attribute for an appliance.

arguments

appliance

One or more appliances

attr

The attribute name that should be removed.

parameters

[attr=*string*]

Can be used in place of the attr argument.

examples

rocks remove appliance attr compute sge

Removes the attribute sge for compute appliances

rocks remove appliance attr compute attr=sge

Same as above.

9.10.3. remove appliance route

rocks remove appliance route {appliance} {address} [address=*string*]

Remove a static route for an appliance type.

arguments

appliance

Appliance name. This argument is required.

address

The address of the static route to remove. This argument is required.

parameters

[address=*string*]

Can be used in place of the 'address' argument.

examples

rocks remove appliance route compute 1.2.3.4

Remove the static route for the 'compute' appliance that has the network address '1.2.3.4'.

9.10.4. remove attr

rocks remove attr {attr} [attr=*string*]

Remove a global attribute.

arguments

attr

The attribute name that should be removed.

parameters

[attr=*string*]

Can be used in place of the attr argument.

examples

rocks remove attr cpus

Removes the global attribute named 'cpus'.

rocks remove attr attr=cpus

Same as above.

9.10.5. remove bootaction

rocks remove bootaction [action=*string*]

Remove a boot action specification from the system.

parameters

[action=*string*]

The label name for the boot action. You can see the boot action label names by executing: 'rocks list bootaction'.

examples

rocks remove bootaction action=os

Remove the 'os' boot action from the system.

9.10.6. remove distribution

rocks remove distribution {distribution}

Remove a distribution specification from the database.

arguments

distribution

Distribution name.

examples

rocks remove distribution rocks-optiputer

Removes the distribution named "rocks-optiputer" from the database.

9.10.7. remove firewall

rocks remove firewall [category=index] {rulename}

Remove a named firewall rule

arguments

[category=index]

[global,os,appliance,host]=index. Specify which version of rulename to remove os=linux, appliance=login, or host=compute-0-0. global, global=, and global=global all refer to the global category. cannot be wildcarded. Specifying just a rulename defaults to the global category

rulename

The particular rule to remove. Cannot be wildcarded

examples

rocks remove firewall global ZZDRACONIAN

Remove the rule named ZZDRACONIAN from the global category

rocks remove firewall appliance=compute MYRULE

Remove the rule named MYRULE from compute appliances

related commands

list firewall

list host firewall

9.10.8. remove host

rocks remove host {host...}

Remove a host from the database. This command will remove all related database rows for each specified host.

arguments

host

List of hosts to remove from the database.

examples

rocks remove host compute-0-0

Remove the compute-0-0 from the database.

9.10.9. remove host alias

rocks remove host alias {host} {name} [name=*string*]

Remove an alias for a host.

arguments

host

One hosts.

name

The alias name that should be removed.

parameters

[name=*string*]

Can be used in place of the name argument.

examples

rocks remove host alias compute-0-0 c-0-0

Removes the alias c-0-0 for host compute-0-0.

rocks remove host alias compute-0-0 name=c-0-0

Same as above.

9.10.10. remove host attr

rocks remove host attr {host} {attr} [attr=*string*]

Remove an attribute for a host.

arguments

host

One or more hosts

attr

The attribute name that should be removed.

parameters

[attr=*string*]

Can be used in place of the attr argument.

examples

rocks remove host attr compute-0-0 cpus

Removes the attribute cpus for host compute-0-0.

rocks remove host attr compute-0-0 attr=cpus

Same as above.

9.10.11. remove host boot

rocks remove host boot {host...}

Removes the boot configuration for a host

arguments

host

One or more named hosts.

examples

rocks remove host boot compute-0-0

Removes the boot configuration for host compute-0-0.

rocks remove host boot compute-0-0 compute-0-1

Removes the boot configuration for hosts compute-0-0 and compute-0-1.

9.10.12. remove host bootflags

rocks remove host bootflags {host...}

Remove the kernel boot flags for a list of hosts.

arguments

host

List of hosts to remove kernel boot flag definitions. If no hosts are listed, then the global definition is removed.

examples

rocks remove host bootflags compute-0-0

Remove the kernel boot flags definition for compute-0-0.

9.10.13. remove host interface

rocks remove host interface {host} {iface} [iface=*string*]

Remove a network interface definition for a host.

arguments

host

One or more named hosts.

iface

Interface that should be removed. This may be a logical interface or the mac address of the interface.

parameters

[iface=*string*]

Can be used in place of the iface argument.

examples

rocks remove host interface compute-0-0 eth1

Removes the interface eth1 on host compute-0-0.

rocks remove host interface compute-0-0 compute-0-1 iface=eth1

Removes the interface eth1 on hosts compute-0-0 and compute-0-1.

9.10.14. remove host key

rocks remove host key {host} [id=*string*]

Remove a public key for a host.

arguments

host

A host name.

parameters

[id=*string*]

The ID of the key you wish to remove. To get the key id, execute: "rocks list host key"

9.10.15. remove host partition

rocks remove host partition {host...} [partition=*string*]

Remove a partition definitions from a host.

arguments

host

A list of one or more host names.

parameters

[partition=*string*]

A single partition to remove from this host. If no partition is specified, then all partitions from the host are removed.

examples

rocks remove host partition compute-0-0

Remove all partitions from compute-0-0.

rocks remove host partition compute-0-0 partition=/export

Remove only the /export partition from compute-0-0.

9.10.16. remove host roll

rocks remove host roll
{host} {name} {version} {arch} [arch=*string*] [name=*string*] [os=*string*] [version=*string*]

Removes a roll assignment to a frontend appliance.

arguments

host

Host name of a frontend machine.

name

Name of the roll (e.g., 'base').

version

Version of the roll (e.g., '5.2').

arch

Architecture of the roll (e.g., 'x86_64').

parameters

[arch=*string*]

Same as 'arch' argument.

[name=*string*]

Same as 'name' argument.

[os=*string*]

The OS version. The default is 'linux'.

[version=*string*]

Same as 'version' argument.

examples

rocks remove host roll frontend-0-0-0 base 5.2 x86_64

Removes the assignment of the roll with name/version/arch of 'base/5.2/x86_64' to frontend-0-0-0.

related commands

list host roll

set host roll

9.10.17. remove host route

rocks remove host route {host} {address} [address=*string*]

Remove a static route for a host.

arguments

host

Name of a host machine. This argument is required.

address

The address of the static route to remove. This argument is required.

parameters

[address=*string*]

Can be used in place of the 'address' argument.

examples

rocks remove host route compute-0-0 1.2.3.4

Remove the static route for the host 'compute-0-0' that has the network address '1.2.3.4'.

9.10.18. remove host sec_attr

rocks remove host sec_attr {host} {attr}

Delete an host specific named attribute from secure attributes table

arguments

host

Host name

attr

The attribute you want to remove

9.10.19. remove network

rocks remove network {network...}

Remove network definition from the system. If there are still nodes defined in the database that are assigned to the network name you are trying to remove, the command will not remove the network definition and print a message saying it cannot remove the network.

arguments

network

One or more network names.

examples

rocks remove network private

Remove network info for the network named 'private'.

9.10.20. remove os

rocks remove os {os}

Remove an OS definition from the system.

arguments

os

The OS type (e.g., "linux", "sunos").

examples

rocks remove os sunos

Removes the OS type "sunos" from the database.

9.10.21. remove os attr

rocks remove os attr {os} {attr} [attr=*string*]

Remove an attribute for an OS.

arguments

os

One or more OS specifications (e.g., 'linux').

attr

The attribute name that should be removed.

parameters

[attr=*string*]

Can be used in place of the attr argument.

examples

rocks remove os attr linux sge

Removes the attribute sge for linux OS machines.

rocks remove os attr linux attr=sge

Same as above.

9.10.22. remove os route

rocks remove os route {os} [address=*string*] [address=*string*]

Remove a static route for an OS type.

arguments

os

The OS type (e.g., 'linux', 'sunos', etc.). This argument is required.

parameters

[address=*string*]

Can be used in place of the 'address' argument.

[address=*string*]

The address of the static route to remove. This argument is required.

examples

rocks remove os route linux 1.2.3.4

Remove the static route for the OS 'linux' that has the network address '1.2.3.4'.

9.10.23. remove roll

rocks remove roll {roll...} [arch=*string*] [version=*string*]

Remove a roll from both the database and filesystem.

arguments

roll

List of rolls. This should be the roll base name (e.g., base, hpc, kernel).

parameters

[arch=*string*]

The architecture of the roll to be removed. If no architecture is supplied, then all architectures will be removed.

[version=*string*]

The version number of the roll to be removed. If no version number is supplied, then all versions of a roll will be removed.

examples

rocks remove roll kernel

Remove all versions and architectures of the kernel roll

rocks remove roll ganglia version=5.0 arch=i386

Remove version 5.0 of the Ganglia roll for i386 nodes

related commands

add roll

create roll

disable roll

enable roll

list roll

9.10.24. remove route

rocks remove route {address} [address=*string*]

Remove a global static route.

arguments

address

The address of the static route to remove. This argument is required.

parameters

[address=*string*]

Can be used in place of the 'address' argument.

examples

rocks remove route 1.2.3.4

Remove the global static route that has the network address '1.2.3.4'.

9.10.25. remove sec_attr

rocks remove sec_attr {attr}

Delete the named attribute from the global secure attributes table

arguments

attr

The attribute you want to remove

9.10.26. remove var

rocks remove var {service} {component} [appliance=*string*] [component=*string*] [service=*string*]

Removes variables in the Rocks key/value database. Variables are defined as a quad of (Appliance, Service, Component, Value). Within a node XML file, values can be retrieved as <var name="Service_Component"/>.

arguments

service

The 'service' of the variable to remove.

component

The 'component' of the variable to remove.

parameters

[appliance=*string*]

If supplied, remove the Service_Component variable that is associated with this appliance.

[component=*string*]

Can be used in place of component argument.

[service=*string*]

Can be used in place of service argument.

examples

rocks remove var service=Condor component=Master

Remove the variable Condor_Master from the database.

rocks remove var service=Condor component=Master appliance=compute

Remove the variable Condor_Master that is associated with the compute appliance.

9.11. report

9.11.1. report bug

rocks report bug

Report info about the system to help debug issues.

examples

$ rocks report bug

Report system info.

9.11.2. report dbhost

rocks report dbhost

9.11.3. report distro

rocks report distro

Output the path prefix for the location of the Rocks distribution.

examples

$ rocks report distro

Output the current path prefix to the distribution.

9.11.4. report host

rocks report host

Report the host to IP address mapping in the form suitable for /etc/hosts.

examples

$ rocks report host

Outputs data for /etc/hosts.

9.11.5. report host attr

rocks report host attr [host] [attr=*string*] [pydict=*bool*]

Report the set of attributes for hosts.

arguments

[host]

Host name of machine

parameters

[attr=*string*]

Output just the value of a particular attribute

[pydict=*bool*]

Output as a python-formatted dictionary. Defaults to false. Only valid if attr parameter is not specified.

examples

$ rocks report host attr compute-0-0

Report the attributes for compute-0-0.

$ rocks report host attr compute-0-0 pydict=true

Report the attributes for compute-0-0 as a python dictionary suitable for input to rocks report script.

$ rocks report host attr compute-0-0 attr=Kickstart_Lang

Output value of the attribute called Kickstart_Lang for node compute-0-0.

related commands

report script

9.11.6. report host bootflags

rocks report host bootflags [host...]

Output the kernel boot flags for a specific host

arguments

[host]

Zero, one or more host names. If no host names are supplied, info about all the known hosts is listed.

examples

$ rocks report host bootflags compute-0-0

Output the kernel boot flags for compute-0-0.

9.11.7. report host config411

rocks report host config411

9.11.8. report host dhcpd

rocks report host dhcpd {host}

Output the DHCP server configuration file for a specific host.

arguments

host

Create a DHCP server configuration for the machine named 'host'. If no host name is supplied, then generate a DHCP configuration file for this host.

examples

$ rocks report host dhcpd frontend-0-0

Output the DHCP server configuration file for frontend-0-0.

9.11.9. report host firewall

rocks report host firewall {host}

Create a report that outputs the firewall rules for a host.

arguments

host

Host name of machine

examples

$ rocks report host firewall compute-0-0

Create a report of the firewall rules for compute-0-0.

9.11.10. report host interface

rocks report host interface {host} [iface=`string`]

Output the network configuration file for a host's interface.

arguments

host

One host name.

parameters

[iface=*string*]

Output a configuration file for this host's interface (e.g. 'eth0'). If no 'iface' parameter is supplied, then configuration files for every interface defined for the host will be output (and each file will be delineated by <file> and </file> tags).

examples

$ rocks report host interface compute-0-0 iface=eth0

Output a network configuration file for compute-0-0's eth0 interface.

9.11.11. report host network

rocks report host network {host}

Outputs the network configuration file for a host (on RHEL-based machines, this is the contents of the file /etc/sysconfig/network).

arguments

host

One host name.

examples

$ rocks report host network compute-0-0

Output the network configuration for compute-0-0.

9.11.12. report host roll

rocks report host roll [host]

Create a report in XML format for a set of rolls that should be applied to a frontend's configuration.

arguments

[host]

Host name of machine

examples

$ rocks report host roll frontend-0-0-0

Report the rolls assigned to frontend-0-0-0.

9.11.13. report host route

rocks report host route {host}

Create a report that contains the static routes for a host.

arguments

host

Host name of machine

examples

$ rocks report host route compute-0-0

Create a report of the static routes assigned to compute-0-0.

9.11.14. report named

rocks report named

Prints the nameserver daemon configuration file for the system.

examples

$ rocks report named

Outputs /etc/named.conf

9.11.15. report resolv

rocks report resolv

Report for /etc/resolv.conf for public side nodes.

examples

$ rocks report resolv

Outputs data for /etc/resolv.conf for the frontend.

9.11.16. report resolv private

rocks report resolv private

Report for /etc/resolv.conf for private side nodes.

examples

$ rocks report resolv private

Outputs data for /etc/resolv.conf for compute nodes.

9.11.17. report script

rocks report script [arch=*string*] [attrs=*string*] [os=*string*]

Take STDIN XML input and create a shell script that can be executed on a host.

parameters

[arch=*string*]

The architecture type.

[attrs=*string*]

Attributes to be used while building the output shell script.

[os=*string*]

The OS type.

examples

$ rocks report host interface compute-0-0 | rocks report script

Take the network interface XML output from 'rocks report host interface' and create a shell script.

9.11.18. report tentakel

rocks report tentakel

Create a report that can be used to configure tentakel.

examples

$ rocks report tentakel

Create a tentakel configuration file.

9.11.19. report version

rocks report version

Output the version of Rocks.

examples

$ rocks report version

Output the current Rocks version.

9.11.20. report zones

rocks report zones

Prints out all the named zone.conf and reverse-zone.conf files in XML. To actually create these files, run the output of the command through "rocks report script"

examples

$ rocks report zones

Prints contents of all the zone config files

$ rocks report zones | rocks report script

Creates zone config files in /var/named

related commands

sync dns

9.12. run

9.12.1. run host

rocks run host
[host...] {command} [collate=*string*] [command=*string*] [delay=*string*] [managed=*boolean*] [num-threads=*string*] [stats=*string*] [timeout=*string*] [x11=*boolean*]

Run a command for each specified host.

arguments

[host]

Zero, one or more host names. If no host names are supplied, the command is run on all 'managed' hosts. By default, all compute nodes are 'managed' nodes. To determine if a host is managed, execute: 'rocks list host attr hostname | grep managed'. If you see output like: 'compute-0-0: managed true', then the host is managed.

command

The command to run on the list of hosts.

parameters

[collate=*string*]

Prepend the hostname to every output line if this parameter is set to 'yes'. Default is 'no'.

[command=*string*]

Can be used in place of the 'command' argument.

[delay=*string*]

Sets the time (in seconds) to delay between each executed command on multiple hosts. For example, if the command is run on two hosts and if the delay is 10, then the command will be executed on host 1, then 10 seconds later, the command will be executed on host 2. Default is '0' (no delay).

[managed=*boolean*]

Run the command only on 'managed' hosts, that is, hosts that generally have an ssh login. Default is 'yes'.

[num-threads=*string*]

The number of threads to start in parallel. If num-threads is 0, then try to run the command in parallel on all hosts. Default is '128'.

[stats=*string*]

Display performance statistics if this parameter is set to 'yes'. Default is 'no'.

[timeout=*string*]

Sets the maximum length of time (in seconds) that the command is allowed to run. Default is '30'.

[x11=*boolean*]

If 'no', disable X11 forwarding when connecting to hosts. Default is 'yes'.

examples

$ rocks run host compute-0-0 command="hostname"

Run the command 'hostname' on compute-0-0.

$ rocks run host compute "ls /tmp"

Run the command 'ls /tmp/' on all compute nodes.

9.12.2. run host sec_attr

rocks run host sec_attr {file}

This command reads a filename that contains pickled information about secure attributes that are relevant to the host. This command is not intended to be run by an administrator.

arguments

file

Filename of the pickled file

9.12.3. run roll

rocks run roll [roll...]

Installs a Roll on the fly

arguments

[roll]

List of rolls. This should be the roll base name (e.g., base, hpc, kernel).

examples

rocks run roll viz

Installs the Viz Roll onto the current system.

9.13. set

9.13.1. set appliance attr

rocks set appliance attr {appliance} {attr} {value} [attr=*string*] [value=*string*]

Sets an attribute to an appliance and sets the associated values

arguments

appliance

Name of appliance

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks set appliance attr compute sge False

Sets the sge attribution to False for compute appliances

rocks set appliance attr compute sge attr=cpus value=2

same as above

related commands

list appliance attr

list host attr

remove appliance attr

remove host attr

set host attr

9.13.2. set attr

rocks set attr {attr} {value} [attr=*string*] [value=*string*]

Sets a global attribute for all nodes

arguments

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks set attr sge False

Sets the sge attribution to False

related commands

list attr

remove attr

9.13.3. set host attr

rocks set host attr {host} {attr} {value} [attr=*string*] [value=*string*]

Sets an attribute to a host and sets the associated values

arguments

host

Host name of machine

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks set host attr compute-0-0 cpus 2

Sets the number of cpus of compute-0-0 to 2

rocks set host attr compute-0-0 attr=cpus value=2

same as above

related commands

list host attr

remove host attr

9.13.4. set host boot

rocks set host boot {host...} [action=*string*]

Set a bootaction for a host. A hosts action can be set to 'install' or to 'os' (also, 'run' is a synonym for 'os').

arguments

host

One or more host names.

parameters

[action=*string*]

The label name for the bootaction. This must be one of: 'os', 'install', or 'run'. If no action is supplied, then only the configuration file for the list of hosts will be rewritten.

examples

rocks set host boot compute-0-0 action=os

On the next boot, compute-0-0 will boot the profile based on its "run action". To see the node's "run action", execute: "rocks list host compute-0-0" and examine the value in the "RUNACTION" column.

9.13.5. set host bootflags

rocks set host bootflags {host...} [flags=*string*]

Set the boot flags for a host. The boot flags will applied to the configuration file that a host uses to boot the running kernel. For example, if a node uses GRUB as its boot loader, the boot flags will part of the 'append' line.

arguments

host

Zero, one or more host names. If no host names are supplied, then the global bootflag will be set.

parameters

[flags=*string*]

The boot flags to set for the host.

examples

rocks set host bootflags compute-0-0 flags="mem=1024M"

Apply the kernel boot flags "mem=1024M" to compute-0-0.

9.13.6. set host comment

rocks set host comment {host...} {comment} [comment=*string*]

Set the comment field for a list of hosts.

arguments

host

One or more host names.

comment

The string to assign to the comment field for each host.

parameters

[comment=*string*]

Can be used in place of the comment argument.

examples

rocks set host comment compute-0-0 "Fast Node"

Sets the comment field to "Fast Node" for compute-0-0.

rocks set host comment compute-0-0 compute-0-1 "Slow Node"

Sets the comment field to "Slow Node" for compute-0-0 and compute-0-1.

rocks set host comment compute-0-0 compute-0-1 comment="Slow Node"

Same as above.

9.13.7. set host cpus

rocks set host cpus {host...} {cpus} [cpus=*string*]

Set the number of CPUs for a list of hosts.

arguments

host

One or more host names.

cpus

The number of CPUs to assign to each host.

parameters

[cpus=*string*]

Can be used in place of the cpus argument.

examples

rocks set host cpus compute-0-0 2

Sets the CPU value to 2 for compute-0-0.

rocks set host cpus compute-0-0 compute-0-1 4

Sets the CPU value to 4 for compute-0-0 and compute-0-1.

rocks set host cpus compute-0-0 compute-0-1 cpus=4

Same as above.

9.13.8. set host installaction

rocks set host installaction {host...} {action} [action=*string*]

Set the install action for a list of hosts.

arguments

host

One or more host names.

action

The install action to assign to each host. To get a list of all actions, execute: "rocks list bootaction".

parameters

[action=*string*]

Can be used in place of the action argument.

examples

rocks set host installaction compute-0-0 install

Sets the install action to "install" for compute-0-0.

rocks set host installaction compute-0-0 compute-0-1 "install i386"

Sets the install action to "install i386" for compute-0-0 and compute-0-1.

rocks set host installaction compute-0-0 compute-0-1 action="install i386"

Same as above.

9.13.9. set host interface channel

rocks set host interface channel {host...} {iface} {channel} [channel=*string*] [iface=*string*]

Sets the channel for a named interface.

arguments

host

One or more hosts.

iface

Interface that should be updated. This may be a logical interface or the MAC address of the interface.

channel

The channel for an interface. Use channel=NULL to clear.

parameters

[channel=*string*]

Can be used in place of the channel argument.

[iface=*string*]

Can be used in place of the iface argument.

examples

rocks set host interface channel compute-0-0 iface=eth1 channel="bond0"

Sets the channel for eth1 to be "bond0" (i.e., it associates eth1 with the bonded interface named "bond0").

rocks set host interface channel compute-0-0 iface=eth1 channel=NULL

Clear the channel entry.

9.13.10. set host interface iface

rocks set host interface iface {host...} {mac} {iface} [iface=*string*] [mac=*string*]

Sets the logical interface of a mac address for particular hosts.

arguments

host

One or more named hosts.

mac

MAC address of the interface whose logical interface will be reassigned

iface

Logical interface.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[mac=*string*]

Can be used in place of the mac argument.

examples

rocks set host interface iface compute-0-0 00:0e:0c:a7:5d:ff eth1

Sets the logical interface of MAC address 00:0e:0c:a7:5d:ff to be eth1

rocks set host interface iface compute-0-0 iface=eth1 mac=00:0e:0c:a7:5d:ff

Same as above.

related commands

add host

9.13.11. set host interface ip

rocks set host interface ip {host} {iface} {ip} [iface=*string*] [ip=*string*]

Sets the IP address for the named interface for one host.

arguments

host

Host name.

iface

Interface that should be updated. This may be a logical interface or the mac address of the interface.

ip

The IP address of the interface. Usually of the form nnn.nnn.nnn.nnn where n is a decimal digit. This format is not enforced. Use IP=NULL to clear.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[ip=*string*]

Can be used in place of the ip argument.

examples

rocks set host interface ip compute-0-0 eth1 192.168.0.10

Sets the IP Address for the eth1 device on host compute-0-0.

rocks set host interface ip compute-0-0 iface=eth1 ip=192.168.0.10

Same as above.

related commands

add host

set host interface iface

set host interface ip

set host interface module

9.13.12. set host interface mac

rocks set host interface mac {host} {iface} {mac} [iface=*string*] [mac=*string*]

Sets the mac address for named interface on host.

arguments

host

Host name.

iface

Interface that should be updated. This may be a logical interface or the mac address of the interface.

mac

The mac address of the interface. Usually of the form dd:dd:dd:dd:dd:dd where d is a hex digit. This format is not enforced. Use mac=NULL to clear the mac address.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[mac=*string*]

Can be used in place of the mac argument.

examples

rocks set host interface mac compute-0-0 eth1 00:0e:0c:a7:5d:ff

Sets the MAC Address for the eth1 device on host compute-0-0.

rocks set host interface mac compute-0-0 iface=eth1 mac=00:0e:0c:a7:5d:ff

Same as above.

rocks set host interface mac compute-0-0 iface=eth1 mac=NULL

clears the mac address from the database

related commands

add host

9.13.13. set host interface module

rocks set host interface module {host...} {iface} {module} [iface=*string*] [module=*string*]

Sets the device module for a named interface. On Linux this will get translated to an entry in /etc/modprobe.conf.

arguments

host

One or more hosts.

iface

Interface that should be updated. This may be a logical interface or the MAC address of the interface.

module

The software device module of interface. Use module=NULL to clear.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[module=*string*]

Can be used in place of the module argument.

examples

rocks set host interface module compute-0-0 eth1 e1000

Sets the device module for eth1 to be e1000 on host compute-0-0.

rocks set host interface module compute-0-0 iface=eth1 module=e1000

Same as above.

rocks set host interface module compute-0-0 iface=eth1 module=NULL

Clear the module entry.

related commands

add host

9.13.14. set host interface name

rocks set host interface name {host} {iface} {name} [iface=*string*] [name=*string*]

Sets the logical name of a network interface on a particular host.

arguments

host

Host name.

iface

Interface that should be updated. This may be a logical interface or the MAC address of the interface.

name

Name of this interface (e.g. newname). This is only the name associated with a certain interface. FQDNs are disallowed. To set the domain or zone for an interface, use the "rocks add network" command, and then associate the interface with the network

parameters

[iface=*string*]

Can be used in place of the iface argument.

[name=*string*]

Can be used in place of the name argument.

examples

rocks set host interface name compute-0-0 eth1 cluster-0-0

Sets the name for the eth1 device on host compute-0-0 to cluster-0-0.zonename. The zone is decided by the subnet that the interface is attached to.

rocks set host interface name compute-0-0 iface=eth1 name=c0-0

Same as above.

related commands

add host

add network

9.13.15. set host interface options

rocks set host interface options {host...} {iface} [iface=*string*] [options=*string*]

Sets the options for a device module for a named interface. On Linux, this will get translated to an entry in /etc/modprobe.conf.

arguments

host

One or more hosts.

iface

Interface that should be updated. This may be a logical interface or the MAC address of the interface.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[options=*string*]

The options for an interface. Use options=NULL to clear. In Rocks 5.4 (linux): options='dhcp' or options='noreport' have special meaning. Bonded interfaces use options field to set up bonding options.

examples

rocks set host interface options compute-0-0 iface=eth1 options="Speed=10"

Sets the option "Speed=10" for eth1 on e1000 on host compute-0-0.

rocks set host interface options compute-0-0 iface=eth1 options=NULL

Clear the options entry.

rocks set host interface options compute-0-0 iface=eth0 options="dhcp"

Linux only: Configure eth0 interface for DHCP instead of static.

rocks set host interface options compute-0-0 iface=eth0 options="noreport"

Linux only: Tell rocks report host interface to ignore this interface when writing configuration files

9.13.16. set host interface subnet

rocks set host interface subnet {host...} {iface} {subnet} [iface=*string*] [subnet=*string*]

Sets the subnet for named interface on one of more hosts.

arguments

host

One or more named hosts.

iface

Interface that should be updated. This may be a logical interface or the MAC address of the interface.

subnet

The subnet address of the interface. This is a named subnet and must be listable by the command 'rocks list network'.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[subnet=*string*]

Can be used in place of the subnet argument.

examples

rocks set host interface subnet compute-0-0 eth1 public

Sets eth1 to be on the public subnet.

rocks set host interface mac compute-0-0 iface=eth1 subnet=public

Same as above.

related commands

add host

9.13.17. set host interface vlan

rocks set host interface vlan {host...} {iface} {vlan} [iface=*string*] [vlan=*string*]

Sets the VLAN ID for an interface on one of more hosts.

arguments

host

One or more named hosts.

iface

Interface that should be updated. This may be a logical interface or the mac address of the interface.

vlan

The VLAN ID that should be updated. This must be an integer and the pair 'subnet/vlan' must be defined in the VLANs table.

parameters

[iface=*string*]

Can be used in place of the iface argument.

[vlan=*string*]

Can be used in place of the vlan argument.

examples

rocks set host interface vlan compute-0-0-0 eth0 3

Sets compute-0-0-0's private interface to VLAN ID 3.

rocks set host interface vlan compute-0-0-0 subnet=eth0 vlan=3

Same as above.

related commands

add host

9.13.18. set host membership

rocks set host membership {host...} {membership} [membership=*string*]

Set the membership for hosts.

arguments

host

One or more host names.

membership

The membership to assign to each host.

parameters

[membership=*string*]

Can be used in place of the membership argument.

examples

rocks set host membership "NAS Appliance" nas-0-0

Sets the membership to 'NAS Appliance' for nas-0-0.

rocks set host membership "NAS Appliance" membership=nas-0-0

Same as above.

rocks set host membership Compute

Sets the membership to 'Compute' for all known hosts.

9.13.19. set host name

rocks set host name {host} {name} [name=*string*]

Rename a host.

arguments

host

The current name of the host.

name

The new name for the host.

parameters

[name=*string*]

Can be used in place of the 'name' argument.

examples

rocks set host name compute-0-0 new-compute-0-0

Changes the name of compute-0-0 to new-compute-0-0.

rocks set host cpus compute-0-0 name=new-compute-0-1

Same as above.

9.13.20. set host power

rocks set host power {host...} [action=*string*] [key=*string*]

Turn the power for a host on or off.

arguments

host

One or more host names.

parameters

[action=*string*]

The power setting. This must be one of 'on', 'off' or 'install'. The 'install' action will turn the power on and force the host to install.

[key=*string*]

A private key that will be used to authenticate the request. This should be a file name that contains the private key.

examples

$ rocks set host power compute-0-0 action=on

Turn on the power for compute-0-0.

9.13.21. set host rack

rocks set host rack {host...} {rack} [rack=*string*]

Set the rack number for a list of hosts.

arguments

host

One or more host names.

rack

The rack number to assign to each host.

parameters

[rack=*string*]

Can be used in place of rack argument.

examples

rocks set host rack compute-2-0 2

Set the rack number to 2 for compute-2-0.

rocks set host rack compute-0-0 compute-0-1 0

Set the rack number to 0 for compute-0-0 and compute-0-1.

rocks set host rack compute-0-0 compute-0-1 rack=0

Same as above.

9.13.22. set host rank

rocks set host rank {host...} {rank} [rank=*string*]

Set the rank number for a list of hosts.

arguments

host

One or more host names.

rank

The rank number to assign to each host.

parameters

[rank=*string*]

Can be used in place of rank argument.

examples

rocks set host rank compute-0-2 2

Set the rank number to 2 for compute-0-2.

rocks set host rank compute-0-0 compute-1-0 0

Set the rank number to 0 for compute-0-0 and compute-1-0.

rocks set host rank compute-0-0 compute-1-0 rank=0

Same as above.

9.13.23. set host roll

rocks set host roll
{host} {name} {version} {arch} [arch=*string*] [name=*string*] [os=*string*] [version=*string*]

Associates a roll with a frontend appliance.

arguments

host

Host name of a frontend machine.

name

Name of the roll (e.g., 'base').

version

Version of the roll (e.g., '5.2').

arch

Architecture of the roll (e.g., 'x86_64').

parameters

[arch=*string*]

Same as 'arch' argument.

[name=*string*]

Same as 'name' argument.

[os=*string*]

The OS version. The default is 'linux'.

[version=*string*]

Same as 'version' argument.

examples

rocks set host roll frontend-0-0-0 base 5.2 x86_64

Associates the roll with name/version/arch of 'base/5.2/x86_64' to frontend-0-0-0.

related commands

list host roll

9.13.24. set host runaction

rocks set host runaction {host...} {action} [action=*string*]

Set the run action for a list of hosts.

arguments

host

One or more host names.

action

The run action to assign to each host. To get a list of all actions, execute: "rocks list bootaction".

parameters

[action=*string*]

Can be used in place of the action argument.

examples

rocks set host runaction compute-0-0 os

Sets the run action to "os" for compute-0-0.

rocks set host runaction compute-0-0 compute-0-1 memtest

Sets the run action to "memtest" for compute-0-0 and compute-0-1.

rocks set host runaction compute-0-0 compute-0-1 action=memtest

Same as above.

9.13.25. set host sec_attr

rocks set host sec_attr {host} [attr=*string*] [crypted=*boolean*] [enc=*string*] [value=*string*]

Set a host-specific secure attribute to the database. The user also has the option of not supplying the value on the command line. The command will prompt the user to enter the secure attribute value, and will not echo this value on screen. If the user chooses to enter the value for the secure attribute by prompt, then the value entered must be in an unencrypted form.

arguments

host

Host name of machine

parameters

[attr=*string*]

Name of the attribute

[crypted=*boolean*]

Is "value" already crypted or not

[enc=*string*]

Encryption scheme to use to crypt the value. Currently supported values are "sha", "crypt", "portable".

[value=*string*]

Value of the attribute

examples

rocks set host sec_attr compute-0-0 attr=db_pw value=DatabasePassword crypted=false enc=sha

Sets a secure attribute called db_pw to the crypted value of "DatabasePassword" using the sha1 encoding schema.

rocks set host sec_attr compute-0-0 attr=db_pw value=77e6674e6d71f898d5fc79424117c86731ca7498 crypted=true

Same as above

9.13.26. set network mtu

rocks set network mtu {network...} {mtu} [mtu=*string*]

Sets the MTU for one or more named networks.

arguments

network

One or more named networks that should have the defined MTU.

mtu

MTU that named networks should have.

parameters

[mtu=*string*]

Can be used in place of 'mtu' argument.

examples

rocks set network mtu optiputer 9000

Sets the "optiputer" MTU address to 9000.

rocks set network mtu optiputer mtu=9000

Same as above.

related commands

add network

set network netmask

9.13.27. set network netmask

rocks set network netmask {network...} {netmask} [netmask=*string*]

Sets the network mask for one or more named networks .

arguments

network

One or more named networks that should have the defined netmask.

netmask

Netmask that named networks should have.

parameters

[netmask=*string*]

Can be used in place of netmask argument.

examples

rocks set network netmask optiputer 255.255.255.0

Sets the netmask for the "optiputer" network to a class-c address space.

rocks set network netmask optiputer netmask=255.255.255.0

Same as above.

rocks set network netmask optiputer cavewave 255.255.0.0

Sets the netmask for the "optiputer" and "cavewave" networks to a class-b address space.

related commands

add network

set network subnet

9.13.28. set network servedns

rocks set network servedns {network} {servedns} [servedns=*bool*]

Sets/Unsets the capability for serving DNS for a given subnet

arguments

network

Name of the Network

servedns

True/False

parameters

[servedns=*bool*]

True/False

9.13.29. set network subnet

rocks set network subnet {network...} {subnet} [subnet=*string*]

Sets the subnet for one or more named networks.

arguments

network

One or more named networks that should have the defined subnet.

subnet

Subnet that named networks should have.

parameters

[subnet=*string*]

Can be used in place of subnet argument.

examples

rocks set network subnet optiputer 132.239.51.0

Sets the "optiputer" subnet address to 132.239.51.0.

rocks set network subnet optiputer subnet=132.239.51.0

Same as above.

rocks set network subnet optiputer cavewave 67.58.32.0

Sets both the "optiputer" and "cavewave" subnet addresses to the same value of 67.58.32.0.

related commands

add network

set network netmask

9.13.30. set network zone

rocks set network zone {network} {zone} [zone=*string*]

Set the zone/domain name associated with a subnet

arguments

network

Network Name

zone

Zone / Domain that the network belongs to. Example: optiputer.net

parameters

[zone=*string*]

Zone / Domain that the network belongs to. Example: optiputer.net

9.13.31. set os attr

rocks set os attr {os} {attr} {value} [attr=*string*] [value=*string*]

Sets an attribute to an os and sets the associated values

arguments

os

Name of os

attr

Name of the attribute

value

Value of the attribute

parameters

[attr=*string*]

same as attr argument

[value=*string*]

same as value argument

examples

rocks set os attr linux sge False

Sets the sge attribution to False for linux nodes

9.13.32. set password

rocks set password

9.13.33. set sec_attr

rocks set sec_attr {attr} [crypted=*boolean*] [enc=*string*] [value=*string*]

Set a secure attribute to the database. The user also has the option of not supplying the value on the command line. The command will prompt the user to enter the secure attribute value, and will not echo this value on screen. If the user chooses to enter the value for the secure attribute by prompt, then the value entered must be in an unencrypted form.

arguments

attr

Name of the attribute

parameters

[crypted=*boolean*]

Is "value" already crypted or not

[enc=*string*]

Encryption scheme to use to crypt the value. Currently supported values are "sha", "crypt", "portable".

[value=*string*]

Value of the attribute

examples

rocks set sec_attr db_pw value=DatabasePassword crypted=false enc=sha

Sets a secure attribute called db_pw to the crypted value of "DatabasePassword" using the sha1 encoding schema.

rocks set sec_attr db_pw value=77e6674e6d71f898d5fc79424117c86731ca7498 crypted=true

Same as above

9.13.34. set var

rocks set var
{service} {component} {value} [appliance=*string*] [component=*string*] [service=*string*] [value=*string*]

Set variables in the Rocks key/value database. Variables are defined as quad of (Appliance, Service, Component, Value). Within a node XML file, values can be retrieved as <var name="Service_Component"/>.

arguments

service

Defines the service name. e.g., service=Kickstart.

component

Defines the component name. e.g., component=PublicDNS.

value

Defines the value for the variable.

parameters

[appliance=*string*]

If supplied, restricts to the named appliance. See 'rocks list appliance' for a listing of appliances.

[component=*string*]

Can be used in place of component argument.

[service=*string*]

Can be used in place of service argument.

[value=*string*]

Can be used in place of value argument.

examples

rocks set var service=Condor component=Master value=localhost

Set the variable name <var name="Condor_Master"/> to 'localhost'.

rocks set var service=Condor component=Master value=localhost appliance=compute

Set the variable name <var name="Condor_Master"/> to 'localhost' and associate it with only compute appliances.

9.14. swap

9.14.1. swap host interface

rocks swap host interface {host} [ifaces=*string*] [sync-config=*boolean*]

Swaps two host interfaces in the database.

arguments

host

Host name of machine

parameters

[ifaces=*string*]

Two comma-separated interface names (e.g., ifaces="eth0,eth1").

[sync-config=*boolean*]

If "yes", then run 'rocks sync config' at the end of the command. The default is: yes.

9.15. sync

9.15.1. sync config

rocks sync config

For each system configuration file controlled by Rocks, first rebuild the configuration file by extracting data from the database, then restart the relevant services.

examples

rocks sync config

Rebuild all configuration files and restart relevant services.

9.15.2. sync dns

rocks sync dns

Rebuild the DNS configuration files, then restart named.

examples

rocks sync dns

Rebuild the DNS configuration files, then restart named.

9.15.3. sync host firewall

rocks sync host firewall

Reconfigure and restart firewall for named hosts.

examples

rocks sync host firewall compute-0-0

Reconfigure and restart the firewall on compute-0-0.

9.15.4. sync host network

rocks sync host network

Reconfigure and restart the network for the named hosts.

examples

rocks sync host network compute-0-0

Reconfigure and restart the network on compute-0-0.

9.15.5. sync host sec_attr

rocks sync host sec_attr {host}

This command syncs the secure attributes of a host.

arguments

host

Hostname(s) whose secure attributes need to be synced.

9.15.6. sync host sharedkey

rocks sync host sharedkey

9.15.7. sync users

rocks sync users

Update all user-related files (e.g., /etc/passwd, /etc/shadow, etc.) on all known hosts. Also, restart autofs on all known hosts.

examples

rocks sync users

Send all user info to all known hosts.

9.16. update

9.16.1. update

rocks update

Download and install updated packages and Rolls from Rocks. This does not include any OS packages. This does not rebuild the distribution or update the backend nodes.

examples

rocks update

Updates the frontend.

Appendix A. Frequently Asked Questions

A.1. Installation

1. Insert-ethers never sees new compute nodes. I also don't see any DHCP messages from compute nodes on the frontend. What is wrong?

Try bypassing the network switch connecting your nodes to the frontend. The swich may be configured to squash broadcast messages from unknown IP addresses, which drops DHCP messages from nodes. To verify your switch is indeed the problem:

1. Connect an ethernet cable between a single compute node and the frontend's "eth0" interface.
2. Install the compute node normally (Install Compute Nodes). You should see the DHCP messages from the node at the frontend.

2. While trying to install a compute node, and when I plug a monitor into the compute node, I see the error message 'Error opening kickstart file /tmp/ks.cfg. No such file or directory' or I see a screen on the compute node asking me to select a language. What went wrong?

A compute node kickstart requires the following services to be running on the frontend:

1. dhcpd
2. httpd
3. mysqld
4. autofs

To check if httpd and mysqld are running:

```
# ps auwx | grep httpd
# ps auwx | grep mysqld
```

If either one is not running, restart them with:

```
# /etc/rc.d/init.d/httpd restart
```

and/or

```
# /etc/rc.d/init.d/mysqld restart
```

The autofs service is called 'automount'. To check if it is running:

```
# ps auwx | grep automount
```

If it isn't, restart it:

```
# /etc/rc.d/init.d/autofs restart
```

Finally, to test if the Rocks installation infrastructure is working:

```
# rocks list host profile compute-0-0
```

This should return a kickstart file.

And to see if there are any errors associated with constructing kickstart files on the frontend:

```
# rocks list host profile compute-0-0 > /dev/null
```

3. I successfully installed all the Rolls, but during the last stage after the machine reboots, the system hangs with the error: *GRUB* Loading Stage2.... What went wrong?

This is an intermittent problem we've seen in the lab as well. The installation is fine, except that the grub installation program, for an unknown reason, did not run correctly.

Here is a workaround:

- Put the Rocks Boot Roll CD in the frontend and boot the frontend.
- At the boot prompt, type:

  ```
  frontend rescue
  ```
- A screen will appear, click the *Continue* button.
- When you see the shell prompt, execute:

  ```
  # chroot /mnt/sysimage
  ```
- Run the grub installation program:

  ```
  # /sbin/grub-install `awk -F= '/^#boot/ { print $2 }' /boot/grub/grub.conf`
  ```

  This should output something similar to:

  ```
  Installation finished. No error reported.
  This is the contents of the device map /boot/grub/device.map.
  Check if this is correct or not. If any of the lines is incorrect,
  fix it and re-run the script `grub-install'.

  # this device map was generated by anaconda
  (fd0)     /dev/fd0
  (hd0)     /dev/hda
  ```
- Exit the chroot environment:

  ```
  # exit
  ```
- Reboot the frontend.
- Take the CD out of the drive and the frontend should come up cleanly.

4. When I try to install a compute node, the error message on the compute node says, "Can't mount /tmp. Please press OK to restart". What should I do?

Most likely, this situation arises due to the size of the disk drive on the compute node. The installation procedure for Rocks formats the disk on the compute node if Rocks has never been installed on the compute node before.

The fix requires changing the way Rocks partitions disk drives. See Partitioning for details.

5. My compute nodes don't have a CD drive and my network cards don't PXE boot, but my compute nodes do have a floppy drive. How can I install the compute nodes?

You will create a boot floppy that emulates the PXE protocol. This is accomplished by going to the web site:

ROM-o-matic.net[1]

Then click on the version number under the *Latest Production Release* (as of this writing, this is version 5.4.3).

Select your device driver in item 1. Keep the default setting in item 2 (Floppy bootable ROM Image). Then click "Get ROM" in item 4.

We suggest using `dd` to copy the downloaded floppy image to the floppy media. For example:

```
# dd if=eb-5.4.0-pcnet32.zdsk of=/dev/fd0
```

Then run `insert-ethers` on your frontend and boot your compute node with the floppy.

A.2. Configuration

1. How do I remove a compute node from the cluster?

On your frontend end, execute:

```
# rocks remove host "[your compute node name]"
```

For example, if the compute node's name is *compute-0-1*, you'd execute:

```
# rocks remove host compute-0-1
# rocks sync config
```

The compute node has been removed from the cluster.

2. Why doesn't startx work on the frontend machine?

Before you can run startx you need to configure XFree86 for your video card. This is done just like on standard Red Hat machines using the `system-config-display` program. If you do not know anything about your video card just select "4MB" of video RAM and 16 bit color 800x600. This video mode should work on any modern VGA card.

3. I can't install compute nodes and I have a Dell Powerconnect 5224 network switch, what can I do?

Here's how to configure your Dell Powerconnect 5224:

You need to set the *edge port* flag for all ports (in some Dell switches is labeled as *fast link*).

First, you'll need to set up an IP address on the switch:

- Plug in the serial cable that came with the switch.

- Connect to the switch over the serial cable.

  The username/password is: admin/admin.

- Assign the switch an IP address:

  ```
  # config
  # interface vlan 1
  # ip address 10.1.2.3 255.0.0.0
  ```

- Now you should be able to access the switch via the ethernet.
- Plug an ethernet cable into the switch and to your laptop.
- Configure the ip address on your laptop to be:

  ```
  IP: 10.20.30.40
  netmask: 255.0.0.0
  ```

- Point your web browser on your laptop to 10.1.2.3
- Username/password is: admin/admin.
- Set the *edge port* flag for all ports. This is found under the menu item: *System->Spanning Tree->Port Settings*.
- Save the configuration.

  This is accomplished by going to *System->Switch->Configuration* and typing 'rocks.cfg' in the last field 'Copy Running Config to File'. In the field above it, you should see 'rocks.cfg' as the 'File Name' in the 'Start-Up Configuration File'.

4. How do I export a new directory from the frontend to all the compute nodes that is accessible under `/home`?

Execute this procedure:

- Add the directory you want to export to the file `/etc/exports`.

  For example, if you want to export the directory `/export/disk1`, add the following to `/etc/exports`:

  ```
  /export/disk1 10.0.0.0/255.0.0.0(rw)
  ```

  This exports the directory only to nodes that are on the internal network (in the above example, the internal network is configured to be `10.0.0.0`)

- Restart NFS:

  ```
  # /etc/rc.d/init.d/nfs restart
  ```

- Add an entry to `/etc/auto.home`.

  For example, say you want `/export/disk1` on the frontend machine (named *frontend-0*) to be mounted as `/home/scratch` on each compute node.

  Add the following entry to `/etc/auto.home`:

  ```
  scratch  frontend-0:/export/disk1
  ```

- Inform 411 of the change:

```
# make -C /var/411
```

Now when you login to any compute node and change your directory to `/home/scratch`, it will be automounted.

5. How do I disable the feature that reinstalls compute nodes after a hard reboot?

When compute nodes experience a *hard* reboot (e.g., when the compute node is reset by pushing the power button or after a power failure), they will reformat the root file system and reinstall their base operating environment.

To disable this feature:

- Login to the frontend
- Create a file that will override the default:

  ```
  # cd /export/rocks/install
  # cp rocks-dist/arch/build/nodes/auto-kickstart.xml \
  site-profiles/5.4.3/nodes/replace-auto-kickstart.xml
  ```

  Where *arch* is "i386" or "x86_64".
- Edit the file `site-profiles/5.4.3/nodes/replace-auto-kickstart.xml`
- Remove the line:

  ```
  <package>rocks-boot-auto<package>
  ```
- Rebuild the distribution:

  ```
  # cd /export/rocks/install
  # rocks create distro
  ```
- Reinstall all your compute nodes

  An alternative to reinstalling all your compute nodes is to login to each compute node and execute:

  ```
  # /etc/rc.d/init.d/rocks-grub stop
  # /sbin/chkconfig --del rocks-grub
  ```

A.3. System Administration

1. How do I use user accounts from an external NIS server on my cluster?

While there is no certain method to do this correctly, if necessary we recommend you use "ypcat" to periodically gather external NIS user accounts on the frontend, and let the default 411 system distribute the information inside the cluster.

The following cron script will collect NIS information from your external network onto the frontend. The login files created here will be automatically distributed to cluster nodes via 411. This code courtesy of Chris Dwan at the University of Minnesota.

```
(in /etc/cron.hourly/get-NIS on frontend)

#!/bin/sh
ypcat -k auto.master > /etc/auto.master
ypcat -k auto.home   > /etc/auto.home
ypcat -k auto.net    > /etc/auto.net
ypcat -k auto.web    > /etc/auto.web

ypcat passwd      > /etc/passwd.nis
cat   /etc/passwd.local /etc/passwd.nis > /etc/passwd.combined
cp    /etc/passwd.combined /etc/passwd

ypcat group       > /etc/group.nis
cat   /etc/group.local /etc/group.nis > /etc/group.combined
cp    /etc/group.combined /etc/group
```

There is no way to insure that UIDs GIDs from NIS will not conflict with those already present in the cluster. You must always be careful that such collisions do not occur, as unpredicatble and undefined behavior will result.

Notes

1. http://www.rom-o-matic.net/

Appendix B. Release Notes

B.1. Release 5.4.3 - changes from 5.4

B.1.1. New Features

- Rocks Security

  The root password supplied during the frontend installation is now used only for the root password of the frontend. Earlier, this password would be used for the MySQL database, Wordpress, and as root passwords for all compute nodes. All these services are now configured randomly-generated passwords.

  To set the root passwords for individual backend nodes, the user can now use command line tool `rocks add host sec_attr ... attr=root_pw` to add a specific root password. `rocks sync host sec_attr ...` for the password to be set. Root passwd on the frontend is changed using the OS-supplied `passwd` command.

  The rationale behind setting random root passwords for all backend nodes is that, if by some means, an attacker gained access to the root account of a backend node, and then the adversary could ran an offline attack against the crypted version of the root password, none of the other nodes would be compromised.

  We've introduced the concept of secure attributes in the database. Any attribute such as passwords, private keys, etc can be stored in the secure attributes table in the database. This table is locked down to be readable only by root, and no one else. It's contents are not transferred during kickstart, and can only be synced using `rocks sync host sec_attr`, which will update the host specific secure attributes. Each attribute must also have a plugin associated with it, to specify the action to be performed on the backend nodes with the secure attribute as its data.

- Changes to 411 infrastructure

  411 files can now only be requested by privileged accounts on backend nodes. This is enforced by checking that 411 requests originate only from privileged ports. Iptables is used to filter out requests that come in from non-privileged ports.

  411 filters now support pre-send, filter, and post-receive functions. These functions are used to modify the contents of the 411 files being distributed, and to perform local system actions once the 411 files have been received and written to disk. These filters are present in the form of 411 plugins which are stored, modified, and enhanced on the frontend.

  The 411 shared key is now distributed outside the kickstart file. This is driven by the command line. The client makes an RPC request to the frontend, and the frontend transfers the 411 shared key out-of-band. It uses rocks command line tools to verify that the request is coming from within the cluster.

- Introduction of the categories and indicies for resolving host specific properties in the database.

  With this release, we've laid the foundation for creating and using random categories of host groups.

  In the previous releases of rocks, the only categories available to system administrators by which they could group hosts were - Global category, Appliance category, OS category, and Hosts category.

  With this release, we've introduced the capability of being able to create user specified categories. Some examples are - rack category, where hosts are grouped by rack, a bio category - where hosts can be grouped by whether or not a set of hosts have the bio roll installed, etc.

We've also introduced the concept of category resolution, where when resolving all the categories that a host can belong to, we can specify an chain of resolution. For example - we can state that compute-0-0 belongs to categories [global, linux, compute, rack0, bio]. In our resolution, we can state that we want the properties of the hosts to be picked up from [global, compute, rack0]. This way compute-0-0 picks up only the properties that are part of its resolution chain of categories.

Since this feature is still prototypical, at the moment, it is used only internally for firewall commands and single resolution chain. Subsequent releases will apply the same technique to attributes and routes.

- Changes to Firewall commands

  The rocks firewall commands now require the presence of a rulename for every iptables rule. These rules are then ordered lexicographically by rule name.

  The firewall command structure has some significant changes to reflect the categories and indicies feature described in the previous bullet point. Please look at the rocks command line documentation for more details on how to run the rocks firewall commands.

- Introduction of Perl and Python rolls

  Two new rolls have been added to Rocks.

  - Perl Roll - The Perl roll contains Perl 5.14.1, and plenty of CPAN modules that are required for the application software in Rocks to function properly. This version of Perl in installed in `/opt/perl`.
  - Python Roll - The version of Python that the core rocks utils depend on, is version 2.4.2. This is a rather dated version of Python. To provide users with the latest versions of python, we've create a Python roll which contains Python version 2.7.2 and version 3.2.1. These are both installed in `/opt/python/`.

B.1.2. Enhancements and Bug Fixes

Base: Updated anaconda to v11.1.2.224

Base: Built against CentOS 5.6 with Updates as of August 7, 2011

Base: database secured during installation. DB security now setup using script.

Base: Only root can create tables in cluster db. Use the rocks.my.cnf file to connect to db as root, because the db is already secured by the time this xml file.

Base: Default random root passwords for client nodes

Base: Added a Development Appliance. This is a backend appliance designed for building rolls without impacting the frontend. The appliance uses the frontend yum repository by default but can be configured to use it own local repository for full isolation.

Base: Removed foundation-perl, cpan, and cpan-support. These have been moved to the Perl roll

Base: Save the debug files from the ramdisk onto /root of the hard disk, so we can use them for post-install analysis.

Base: Properly report disk partitions into the database for software RAID file systems. Increase installation speeds for clients with software RAID file systems.

Base: 411 shared key is no longer transferred through kickstart. It is transferred through "rocks sync host sharedkey" 411 configuration is now generated through "rocks report host config411" 411 files are not transferred during kickstart. They are now transferred at first boot.

Base: Unambiguous add host command. Previously it was not possible add hostnames that were command line actions. For example, you could not add a host called "attr" because the command "rocks add host attr" exists.

Base: Remove shadow attributes from attributes tables and added secure attributes tables.

Base: Set primary interface of login servers back to private. Otherwise SGE behaves badly.

Base: Firewall rules now have rulenames and lexical ordering.

Base: Add profile.d/ssh-keygen to login appliances

Base: Hard link /etc/ssh/authorized_keys/id_rsa.pub to /root/.ssh/id_rsa.pub. If a user (or update) sets root's directory permissions tightly, we still read the public key.

Base: Block non-priviledged traffic to 411 port from all networks, including localhost.

Base: Add variable to manage number parallel instances of "rocks sync host" commands.

Base: Support for pre-send, filtering, and post receive actions in 411.

Base: Minor modification to 411put. Use a get_filename function instead of a filename constant.

Base: 411 plugins can access host attributes

Base: 411 filters user password, and shadow information of users with UID < 500

Base: Initial Support for host categories and indicies.

Base: named.conf bug fixes - Now supports multiple subnets on non-octet boundaries.

Base: If yum install fails due to dependency error, force install using rpm --nodeps.

Base: Rocks run roll now honors the "--interpreter" flag to the post sections.

Base: Honor .<arch> directive to yum install. When installing packages use, "yum install <package>" instead of "yum install <packagefile>.rpm"

Base: Use YUM instead of RPM for rocks run roll This fixes two issues - On 64bit we were not installing the 32bit RPMs, and name.arch packages were not being installed.

Base: Support for HVM when using the Xen roll.

Base: Runtime optimization: Do not regenerate ALL pxeboot files. Just those for the hosts specified on the command line.

Base: Parallel class now takes care of serializing tasks that may overrun the system. If more than a set number of tasks are running, then requesting tasks will wait till slots are available to run Each task now prints out error messages if they fail on remote hosts. This way, we can track which syncs failed and which ones succeeded.

Base: Root ssh key needs to be passwordless to allow command/sync access to backend nodes. If root, create ssh key without passphrase. If normal user, create key interactively.

Base: Updated Java to 1.6 update 26

Base: Support for versioned centrals

HPC: Removed ganglia-web-frontend-addons from HPC roll

HPC: Update IOZone to 3.397

HPC: Update MPICH2 to v1.4

HPC: Update OpenMPI to v1.4.3

Ganglia: Updated to v3.2.0

Ganglia: gmond.conf cleaned to support updated version.

Ganglia: Updated apr to v1.4.5

Ganglia: Updated apr-utils to v1.3.13

Java: Tomcat-connectors rpm bug fix. Now no longer generates conflicts when installing tomcat httpd configuration.

SGE: Upgraded SGE to Open Grid Scheduler v6.2 update 5p2

SGE: Move the login appliance configuration out of the SGE roll and into the Base roll.

Web Server: Updated Wordpress to 3.1.3

Web Server: Updated Rocks Theme for Wordpress.

Web Server: Scrub root password from the installation and set the admin password to a string that never hashes.

Web Server: Wordpress admin password can be reset only if valid admin email is supplied.

Xen: Now support HVM as well as paravirtual instances

Xen: no longer need 'rocks-create-vlan'

Xen: added a report to create the xendomains configuration file

Xen: save the CA key and CA certificate that are used to authenticate

libvirt messages. Xen: touch /var/lock/subsys/xendomains in order to save running VMs

Xen: Ability to put frontend on arbitrary vm-container and set its name

Xen: Explicitly state default for virtualization type

Area51: set the right attr for the default tripwire email address

Area51: send tripwire reports to multiple recipients

Bio: Login appliance gets Bio Roll

Bio: Biopython now depends on Python Roll

Bio: BioPython upgraded to v1.5.7

Bio: BioPerl now depends on Perl Roll

Bio: BioPerl CPAN support utils updated

Bio: All BioPerl CPAN utils now built with cpan2dist

Bio: Updated EMBOSS to v6.3.1

Bio: Updated Autodock suite to v4.2.3

Bio: Update CGView to v2.0

Bio: Update fasta to v36.3.5a

Bio: Update MrBayes to v3.1.2

Bio: Update Blast to v2.2.25

Bio: Update reportlab to v2.5

Bio: Update t_coffee to v8.99

Bio: Update WGS to v6.1

Condor: Update Condor to v7.6.2

Condor: rocks login appliance submits jobs to condor

Condor: Experimental: Support submission to EC2

Condor: Add RANK parameter

B.2. Release 5.4 - changes from 5.3

B.2.1. New Features

- Redesign of the Avalanche Installer.

  While observing the performance of the Avalanche Installer on a 1000-node machine, it became obvious that we must reduce as much traffic to the frontend as possible. This led to replacing the python BitTorrent-based installer with a BitTorrent-inspired installer written in C. The C code allows us to put more files into the peer-to-peer network, most notably: product.img (160KB), stage2.img (108MB) and updates.img (98MB).

  To further reduce traffic to the frontend, the frontend now sends package predictions to installing nodes. When a node asks for a package, the tracker on the frontend sends a list of node addresses where that package can be found, plus a list of the next 9 packages that node will most likely ask for next. When similar appliances are concurrently installing, this reduces tracker traffic by 10x.

  Installing nodes can be grouped. When an installing node asks the tracker for the location of a package and if other nodes are concurrently installing, the tracker will favor nodes that are in the same group as the requesting node, that is, the list the tracker sends back to the installing node will have nodes from the same group as the installing node at the top of the list. The default grouping is by rack, but it can be controlled by the "coop" attribute. For example, if you would like to put all nodes from rack 0 and rack 1 in the same group (named "red"), you would execute: "rocks set host attr rack0 rack1 coop red".

  One can specify multiple trackers and multiple "package servers". A package server is a node that is "guaranteed" to have the requested file (e.g., the frontend).

  For every downloaded file, an MD5 checksum verification is performed. This detects the case where a peer may have corrupted a file and prevents the corrupted file from spreading into the peer-to-peer network.

- Channel bonding for nodes is now controlled by the Rocks command line.

  Channel bonding configuration for a node is stored in the database and can be added, removed or modified with the Rocks command line (e.g., "rocks add host bonded ..."). After channel bonding is configured for a node, it can be dynamically applied by executing "rocks sync host network ...".

- All nodes’ firewall rules are controlled by the Rocks command line.

  The rules for all the nodes are stored in the database and can be added, removed or modified with the Rocks command line (e.g., "rocks open host firewall", "rocks close host firewall", "rocks remove host firewall"). After a node’s firewall settings are changed, they can be applied to the node on-the-fly with "rocks sync host firewall ’hostname’" (this command is also called when the user executes "rocks sync host network ...").

- Introduction of "Air Traffic Control".

We've developed a service known as the "Airboss" that resides on the physical frontend (in Dom0) and it allows non-root users to control their VMs. The motivation for this service is that libvirt (a virtualization API written by RedHat that can control several different virtualization implementations) assumes "root" access to control and monitor VMs.

The Airboss in Rocks is a small service that uses digitally signed messages to give non-root users access to their virtual cluster (and only their virtual cluster). The Airboss relies upon public/private key pairs to validate messages. The administrator of the physical hosting cluster must issue a single command to associate a public key with a particular virtual cluster. At that point, the full process of powering up, powering down and installing a virtual cluster can be controlled by the (authorized) non-root user.

In addition to VM power control, we've also added the ability to attach to a VM's console. This allows users to see the entire boot sequence for a VM starting from the "BIOS" boot messages.

Several Rocks commands were added to support this feature: "rocks create keys" (to create public/private key pairs), "rocks set host power" (to power up/down VMs and to forcibly install a VM, akin to PXE booting a physical machine), and "rocks open host console" (to attach to a VM's console).

- "greceptor" replaced with "channeld".

  The wire protocol for Ganglia messages changed which required a major overhaul to greceptor. We made the decision to write a simple RPC-based service (named 'channeld') to take over the responsibilities of greceptor. Channeld accepts 411-put requests and acts on them by using 411-get to download files under the control of 411.

  All other components of 411 remain unchanged, only the notification engine has been enhanced.

- DNS resolution for multiple domains.

  The DNS naming system on the frontend now supports multiple zones, where each subnet managed by the frontend can be put into a different zone. The DNS service can be turned on or off for each individual zone.

- Login appliance support.

  A node can be configured as a Login appliance. By default, a Login appliance can submit jobs, but it cannot execute jobs.

- Set the name of a host based on the name of a specific network interface.

  The "primary_net" attribute allows nodes to have /bin/hostname set to the name of a network interface other than "private". This is useful for login or other multiple interface appliances.

- Easily swap 2 interfaces with one Rocks command.

  To swap the settings of 2 interfaces, execute "rocks swap host interface ...".

- Created a GIT repository for Rocks-related source code.

  The host "git.rocksclusters.org" is a GIT repository for all core Rocks code, UCSD Triton Resource code and Rocks contrib code.

B.2.2. Enhancements

OS: Based on CentOS release 5/update 5 and all updates as of November 2, 2010.

Base: Anaconda installer updated to v11.1.2.209.

Base : no longer remap the private network to "eth0", instead Rocks keeps track of the network a node kickstarted from and maps that network to the "private" network. For example, if a node kickstarted off "eth1", then "eth1" will be mapped to the private network.

Base : hardened the Anaconda installer to more aggressively write the grub configuration files onto the boot disk. This helps to mitigate the "hang while trying to load Grub stage2" issue.

Base : removed ext4 kernel module from installation environment. We found that trying to mount a swap partition as an ext4 file system frequently caused kernel panics during installations.

Base : added ksdevice=bootif to all the PXE boot targets. This improves installation speed by reusing the IP address/interface information when a node PXE boots. Previously, a node would re-scan all ethernet interfaces.

Base : when a node XML file has a syntax error, "rocks list host profile" prints out the name of the node XML file and the line number where the syntax error occurred.

Base : "rocks run host" now spawns multiple parallel threads when multiple hosts are supplied. Also added the following parameters: timeout (thanks Tim Carlson!), delay, stats, collate and num-threads.

Base : yum configuration default modified to bind to the frontend's public IP instead of the private. This facilitates easy package installation for external nodes (e.g., nodes running on a public cloud).

Base : non-existent attributes are considered to be false conditionals when building configuration files.

Base : "precedes" method added for Rocks command plugins to enable fine-grained ordering of plugin execution.

Base : network interfaces under Linux support 2 new specific modes: "dhcp" and "noreport". The "dhcp" mode indicates that the interface should always DHCP to get its address. The "noreport" mode specifies that no "ifcfg-*" file should be written for the interface. If a mode is not specified for an interface, then Rocks will create an "ifcfg-*" file for the interface based on values set in the database (just like it did in the previous release).

Base : IPMI now uses the interface channel column in the networks table to specify the baseboard controller channel number.

Base : text inside "changelog" tags is now wrapped in CDATA to allow XML escape characters. This is only supported for node XML files found within Rolls (not for node XML files found under /export/rocks/install/site-profiles.

Base : rolls can be built without a complete copy of the Rocks source code. They use the Rocks development environment found under /opt/rocks/share/devel on a frontend.

Area51: tripwire updated to v2.4.2.

Bio: refreshed CPAN modules.

Bio: refreshed CPAN MPI-Blast.

Bio: added Celera Whole Genome Sequence Assembler.

Condor: updated to v7.4.4.

Condor: automated Condor configuration completely retooled: 1) the configuration is Rocks command based instead of standalone CondorConf tool, 2) it supports dynamic update of any/all configurations on nodes, 3) it uses Rocks command plugins to allow additional automated condor config (e.g., via plugin, it can turn on MPI support).

Condor: supports a pool password (shared secret) for additional host verification.

Condor: integrates with EC2 roll to extend Condor pools with EC2 Hosts.

Condor: support added for port ranges to facilitate firewall configuration.

Condor: local copy of Condor's manpages added to roll documents.

Condor: support for updating Condor on nodes without re-installation (e.g., rocks run host "yum update condor" ; rocks sync host condor).

Ganglia: monitor-core updated to v3.1.7.

Ganglia: rrdtool updated to v1.4.4.

Ganglia: the Ganglia Roll can now be added on-the-fly to an existing frontend.

Ganglia: all nodes send out their metric metadata every 3 minutes. In the past, when gmond was restarted on the frontend, it couldn't collect metrics from the nodes because it had no metadata from the nodes (and it didn't have a way to ask the nodes because the nodes are configured in "deaf" mode).

HPC: iozone updated to v3.347.

HPC: iperf updated to v2.0.5.

HPC: MPICH2 updated to v1.2.1p1.

HPC: OpenMPI updated to v1.4.3.

HPC: rocks-openmpi is the default MPI and it is configured with mpi-selector.

SGE: SGE updated to V62u5.

SGE: any host can be configured to be an execution host by setting the host's "exec_host" and "sge" attributes to true and any host can become a submission host by setting the host's "submit_host" and "sge" attributes to true.

Web-server: mediawiki updated to v1.16.0.

Web-server: wordpress updated to v3.0.1.

Xen: any node can how host Xen virtual machines. This is controlled with the "xen" attribute.

Xen: set the power for all nodes in a virtual cluster (except the VM frontend) with one command ("rocks set cluster power ..."). Power settings can be "on", "off" or "install" (turn on and force installation).

Xen: allow virtual machines to define VLAN tagged interfaces. Previously, VLAN tagging was only supported for physical interfaces.

B.2.3. Bug Fixes

Base: non-root users can no longer see the encrypted passwords with 'rocks list host attr'. Hashed passwords are now stored in a 'shadow' column in the attribute tables.

Base: the "%" in "rocks run host %" now returns all hosts. Thanks to Tom Rockwell for the fix.

Base: If an ethernet switch sends out a DHCP request, the DHCP server no longer sends it the "filename" and "next server" in the DHCP response. This caused some switches not to properly load their firmware. More generally, this is controlled by the "kickstartable", "dhcp_filename" and "dhcp_nextserver" attributes.

Base: "rocks set password" asks the user to confirm their new password.

Base: when a node requests a kickstart file and if the frontend determines that the frontend is too "busy", the kickstarting node now correctly does a random backoff before re-requesting its kickstart file. Prior to this fix, a node would backoff for 30 seconds.

Base: multiple conditionals can now be present in XML tags.

Base: fixed a graph traversal issue. In the past, if you had the graph "a" (cond) to "b" to "c" and if "cond" was false, the graph traversal would include "a" and "c". Now it just includes "a".

Base: permissions set in the "file" tag are preserved even if there are other "file" tags for the same file that don't set the file's permissions. The bug was when a later "file" tag without a "perms" attribute was encountered, the file's permissions were cleared.

Base: "file" tags now support "os" conditionals.

Base: in insert-ethers, appliances that are marked "not kickstartable" will not have to wait for a kickstart file. In the past, one had to hit the "F9" (force quit) key to exit insert-ethers when discovering non kickstartable appliances (e.g., ethernet switches).

Base: IPMI configuration cleaned up. Rocks no longer generates erroneous entries in modprobe.conf or /etc/sysconfig/ifcfg-ipmi.

Base: The "pre" tag now supports the "interpreter=" attribute.

Bio: eliminated "Permission Denied" errors during multiple runs on the same BLAST database by different users.

SGE: made the job collection metric more efficient. Previously, when 100's of jobs are submitted to a frontend's queue, the SGE metric would take so long to execute, it caused gmond to stop gathering metrics for all hosts.

SGE: the number of CPUs array jobs consume are now correctly counted.

B.3. Release 5.3 - changes from 5.2

B.3.1. Enhancements

OS: Based on CentOS release 5/update 4 and all updates as of December 15, 2009.

Area51: chkrootkit updated to v0.49.

Base: Anaconda installer updated to v11.1.2.195.

Base: Added support HP's Smart Array controllers (e.g., the 'cciss' driver) in the Rocks partitioning code.

Base: Moved JDK to the Base Roll (the Java Roll has been discontinued).

Base: Added the 'ssh_use_dns' attribute (default setting is 'true'). If set to 'false', then DNS is not consulted when trying to make an SSH connection. This speeds up SSH connections for frontend's that don't have access to a functioning DNS server.

Base: Added the 'Kickstart_DefaultLeaseTime' and 'Kickstart_MaxLeaseTime' attributes to give the user the ability to control their own DHCP lease timeouts.

Base: Added IMPI tools and configuration utilities.

Base: Updated foundation-perl to v5.10.1.

Base: Updated foundation-perl-Tk to v804.028.

Base: Experimental CPAN support. This introduces a utility that lets the user create Rocks RPMs directly from CPAN along with all dependencies.

Bio: Updated Emboss to v6.1.0.

Bio: Updated biopython to v1.52.

Bio: Updated clustalw to v2.0.12.

Bio: Updated fasta to v35.4.9.

Bio: Updated fftw to v3.2.2.

Bio: Updated gromacs to v4.0.5.

Bio: Updated iolib to v1.12.1.

Bio: Updated phylip to v3.69.

Bio: Updated t_coffee to v8.14.

Ganglia: Updated rrdtool to v1.3.8.

HPC: Updated iozone to v3.291.

HPC: Updated iperf to v2.0.4.

HPC: Updated MPICH2 to v1.1.1p1.

HPC: Updated OpenMPI to v1.3.3.

SGE: Updated to v6.2 update 4.

Viz: Updated nVidia driver to v190.42.

Web-server: Updated mediawiki to v1.15.1.

Web-server: Updated wordpress to v2.8.4.

B.3.2. Bug Fixes

Area51: tripwire is now correctly configured on the frontend.

Base: Increase the amount of memory PHP can use from 64 MB to 256 MB. This is essential for viewing Ganglia data for large clusters.

Base: During installation, don't copy over the rocks-cdrom block device file. On most systems, there is no issue, but there are some systems where this code would copy over the entire contents of the CD/DVD if the CD/DVD was still in the tray (which is shouldn't be, because after the rolls are copied to the frontend, the CD/DVD is umounted and ejected).

Base: Set the '--utc' flag when hwclock is called. Without the flag, the localtime is written to the hardware clock. Then, at first boot, the system reads the hardware clock respecting the UTC settings in /etc/sysconfig/clock. So the system time ends up being off by the offset to UTC.

Base: On first boot, Set the number of CPUs for the frontend in the database to actual number of detected CPUs. In the past, this value was hard-coded to 1.

Base: Fix for frontend's that use DNS servers that are configured to return a 'catch all' address for non-existant domain names. Do a reverse lookup to get the IP address. In the previous release, one would see an 'XXX' entry for the IP address of the frontend in /etc/hosts.

Base: Put double quotes around 'option domain-name', otherwise, if the domain name was structured like xxx.yyy, then the DHCP service would not start.

Base: For tentakel, throttle the number of concurrent connections to 100 (useful for large clusters).

Base: Rewrite the pxelinux.cfg files after setting the run/install action in the nodes table (e.g., 'rocks set host installaction ...').

Base: For 'rocks set password', make sure password changing code accesses the Rocks foundation database.

Base: For 'rocks sync host network', make sure to update the static routes file.

Base: Fixed 'insert-ethers --replace'. Before, it correctly removed the node, but then readded it at 'the end' of namespace, that is, it forgot the rack/rank of the replaced node.

Base: Make tentakel more resilient to hanging nodes.

Ganglia: Fix to save all RRD data on frontend shutdown.

Ganglia: Fixes to physical views. Make sure all nodes are positioned in rack/rank order.

Ganglia: Fixes to rediscover all backend nodes after a frontend reboot. Without this fix, if gmond was restarted on the frontend, then it requires restarting gmond on all the compute nodes in order to get metrics to report again.

Kernel: Fixes to preserve the timestamp from RPMs on Rolls that are installed via CD/DVD. Prior to this fix, all RPMS in Rolls that were installed from CD/DVD would have the timestamp of the time when the frontend was installed. This causes issues when trying to update RPMs after the frontend was installed.

SGE: When a host is removed (e.g., 'rocks remove host ...'), make sure the queue associated with that node is also removed.

SGE: Build the DRMAA library with Java support.

Xen: Respect the MTU value of the physical interface when bringing up Xen bridges.

Xen: Fix to enable physical devices (partitions, LVM partitions, etc.) to be used as devices for virtual disks.

Xen: Fix to 'ip' address flag in 'rocks add host vm'. Without the fix, if an IP address was specified on the command line, the command would throw an exception.

Xen: Fix to support for virtual compute nodes that are managed by a physical frontend.

B.4. Release 5.2 - changes from 5.1

B.4.1. New Features

- Solaris support for client nodes

  With the new JumpStart Roll, one can now install and configure a Linux-based Rocks frontend to "JumpStart" Solaris-based back-end machines.

- Attributes

  Can assign "attributes" to nodes at four levels: global, appliance type, OS (e.g., Linux or SunOS), and host. An attribute can be accessed in an XML node as an entity. For example, if you assign the attribute "foo" with the value "123" to compute-0-0 (i.e., with the command, "rocks set host attr foo 123"), then in an XML node file, you can access the value of the attribute foo with "&foo;".

Attributes also enable "conditionals". Using the example above, a "post" section can be optionally executed based on the value of an attribute. For example, if a post section is defined as: <post cond="foo='123'"> then this post section will only be executed if the attribute "foo" is set to "123" for the installing host.

B.4.2. Enhancements

OS: Based on CentOS release 5/update 3 and all updates as of June 22, 2009.

Base: Anaconda installer updated to v11.1.2.168.

Base: Isolated MySQL for the Rocks database under /opt/rocks.

Base: Converted all 'dbreports' to the Rocks command line.

Base: Configure 'MTU' for networks through the Rocks command line.

Base: Configure network routes through the Rocks command line.

Base: Configure host aliases through the Rocks command line.

Base: Added '/var/log/authpriv' to log rotate list.

Base: Added 'iburst' flag to NTP configuration on frontend.

Base: Added version and release info to vmlinuz and initrd.img to enable cross-kickstarting via PXE and making it easier to host different Rocks VMs on a physical system.

Base: Added a YUM configuration file to each node that points to the distribution on the frontend.

Bio: Updated biopython to v1.50.

Bio: Updated clustalw to v2.0.11.

Bio: Updated fasta to v35.4.7.

Bio: Updated fftw to v3.2.1.

Bio: Updated elph archive in glimmer to v1.0.1.

Bio: Updated gromacs to v4.0.1.

Bio: Updated perl-Data-Stag to v0.11.

Bio: Updated perl-Digest-MD5 to v2.38.

Bio: Updated perl-File-Temp to v0.21.

Bio: Updated perl-GD to v2.41.

Bio: Updated perl-GD-SVG to v0.33.

Bio: Updated perl-Graph to v0.91.

Bio: Updated perl-HTML-Parser to v3.60.

Bio: Updated perl-HTML-Tagset to v3.20.

Bio: Updated perl-PathTools to v3.30.

Bio: Updated perl-SOAP-Lite to v0.710.08.

Bio: Updated perl-SVG to v2.49.

Bio: Updated perl-SVG-Graph to v0.02.

Bio: Updated perl-Scalar-List-Utils to v1.21.

Bio: Updated perl-Storable to v2.20.

Bio: Updated perl-Text-Iconv to v1.7.

Bio: Updated perl-URI to v1.38.

Bio: Updated perl-XML-Parser to v2.36.

Bio: Updated perl-XML-Twig to v3.32.

Bio: Updated perl-XML-Writer to v0.606.

Bio: Updated perl-bioperl to v1.6.0.

Bio: Updated perl-libnet to v1.22.

Bio: Updated perl-libwww-perl to v5.826.

Bio: Updated phylip to v3.68.

Bio: Updated reportlab to v2.3.

Bio: Updated t_coffee to v7.81.

Ganglia: Updated to v3.1.2.

HPC: Updated OpenMPI to v1.3.2.

HPC: Updated MPICH2 to v1.0.8p1.

HPC: Updated stream to v5.9.

Java: Updated java to v1.6.0_13.

Java: Updated jdk to v6 update 13.

Java: Updated antlr to v3.1.

Java: Updated jboss to v5.0.1.GA.

Java: Updated jogl to v1.1.1.

SGE: Updated to v6.2 update 1.

Viz: Support for single, dual and quad display nodes.

Viz: Chromium support for 32-bit and 64-bit applications.

Viz: Added CUDA driver. Users can optionally run CUDA programs on the tiled-display nodes.

Viz: User updatable nVidia driver. Makes it easy for users to refresh the nVidia driver without having to wait for an updated Viz Roll.

Xen: Using libvirt instead of 'xm' command line programs to start/stop VMs.

Xen: Allow VM disk to be backed by a physical disk partition.

Xen: Use threading in the 'rocks add cluster' command to decrease the time to add a virtual cluster.

B.4.3. Bug Fixes

Base: Fix for software RAID partitioning.

Base: Increase timeout for package downloads when on slow networks (e.g., 100 or 10 Mbit).

Xen: Fix to ensure all routes are active after Xen is started.

Xen: Fix ’rocks set host vm’ command to allow users to resize a VM’s disk.

B.5. Release 5.1 - changes from 5.0

B.5.1. New Features

- Support for Virtual Clusters

  Virtual frontends and virtual compute nodes are now supported. The network for a VM frontend its VM compute nodes are contained within its own VLAN.

  A virtual cluster is added with "rocks add cluster fqdn=X ip=Y num-computes=Z". See "rocks add cluster help" for details.

- Can build rolls outside of Rocks source tree.

  All roll building support files are under `/opt/rocks/share/devel`.

- Can reconfigure a compute node’s network without rebooting.

  Rocks commands were added to support this. See the documentation for the procedure.

- Distribution moved to `/export/rocks/install`.

  No longer require NFS on the frontend to properly host a Rocks Distribution. This will make moving user accounts to an external NFS server easier.

- Fine-grained control over the "boot" and "install" kernel for Xen VMs.

  Rocks commands where added to support this feature. For details, execute: "rocks help installprofile" and "rocks help bootprofile".

B.5.2. Enhancements

OS: Based on CentOS release 5/update 2 and all updates as of November 4, 2008.

Base: Anaconda installer updated to v11.1.2.113.

Base: Increased the / partition default size to 16 GB.

Base: Opened the ’www’ and ’https’ ports to the local public network.

Base: In Avalanche Installer, added code to check if a package is requested twice in a row. If it is, we assume the package is corrupted. In this case, we toss the package and retrieve it the package from the frontend.

Base: Added Rocks commands to manage the "aliases" table.

Base: Added "rocks remove roll" command. Thanks to Brandon Davidson from the University of Oregon for the code.

Base: The command "rocks-dist" is replaced with "rocks create distro".

Base: Disabled the watchdog for frontend installs that boot off a CD/DVD.

Base: Changed boot command from "frontend" to "build". To build a frontend, when you see the "boot:" prompt, now type: "build".

Web Server: Wordpress updated to v2.6.1.

Web Server: Updated Wordpress theme.

Area51: All commands converted to Rocks command line.

HPC: Updated OpenMPI to v1.2.7.

HPC: Updated MPICH2 to v1.0.7.

Java: Fixed a bug in the graph that Java from properly installing on compute nodes.

Restore: All files under /export/rocks/install/contrib are now included in the Restore Roll.

Restore: All files in /var/named/*local are now included in the Restore Roll.

Restore: The frontend's ssh machine keys are now included in the Restore Roll.

SGE: Updated to v6.1 update 5.

SGE: Added a script to reinstall a cluster by submitting an SGE job.

B.5.3. Bug Fixes

Base: Reverse domain lookups now work for subnets that don't fall on an octet boundary.

Base: Bootflags now carry over between reinstallations and reboots. In the previous release, a reboot would "forget" the bootflags set by the user with the Rocks command line.

Base: Added full path to "mksquashfs". Now can build distribution when using "sudo".

Restore: Ethernet Switches, Power Units and Remote Management appliances are now properly saved in the Restore Roll.

Ganglia: Fixed the 'tail +4' bug in the cron job.

SGE: Fixed the display in the "Job Queue" on the frontend's web site. SGE now reports the correct number of CPUs in use.

B.6. Release 4.3 - changes from 4.2.1

B.6.1. New Features

- Rocks Command Line

Initial release of the Rocks command line which facilitates non-SQL administrative access to the database. All Rocks commands have a regular structure of "rocks <verb> <component>". For example, to list all hosts that have been discovered by the frontend, execute: "rocks list host".

All rocks commands can be listed by executing: rocks. Also, help is included with each command. For example, for help on the command "rocks add host", execute: "rocks add host help".

For an overview of the Rocks commmand line, see Introduction to the Rocks Command Line[1]. The reference for all Rocks commands can be found here[2].

- PXE First

  Hosts can now be configured in BIOS with a boot order of CD, PXE, Hard Disk (previous releases of Rocks required: CD, Hard Disk, PXE). In combination with the Rocks command line, node-specific installation parameters are easily supported. For details on PXE First, see Boot Order and PXE First[3].

  Note: The boot order of (CD, HD, PXE) continues to be supported in Rocks 4.3. That is, existing Rocks clusters can be upgraded without requiring the cluster owner to change any BIOS settings.

B.6.2. Enhancements

OS: Based on CentOS release 4/update 5 and all updates as of July 4, 2007.

Base: Anaconda installer updated to v10.1.1.63.

Base: Performance improvement when building torrent files for the Avalanche Installer.

Base: Database indirects. More flexibility with Rocks variables.

Grid: Globus updated to gt4.0.4 with web services.

Condor: updated to v6.8.5.

PVFS2: updated to v2.6.3.

Java: updated to v1.5.0_10.

Ganglia: updated to v3.0.4.

HPC: Now using OpenMPI and PVM from RedHat distribution.

B.6.3. Bug Fixes

Base: Install now supports machines which have more than 26 disk drives.

Base: 411 clients now atomically update files.

Condor: Max heap size properly set for java programs on small and large memory machines.

Condor: All logging written to /var/opt/condor.

B.7. Release 3.2.0 - changes from 3.1.0

New Feature - Added the Condor Roll. This brings the distributed high-throughput features from the Condor project to Rocks clusters.

New Feature - Added the Area51 Roll. This roll contains security tools and services to check the integrity of the files and operating system on your cluster.

New Feature - Ganglia RSS news event service.

Enhancement - Improved network handling for compute nodes: any interface may be used for the cluster private network, not simply the default "eth0".

Enhancement - Better support for cross-architecture clusters containing x86 and x86_64 machines.

Enhancement - GM device driver now builds and loads on compute nodes that have a custom kernel (e.g., a kernel from kernel.org).

Enhancement - Software RAID for custom compute node partitioning is supported.

Enhancement - Added variables for root and swap partition. If you only want to change the size of root and/or swap, you only have to reassign two XML variables.

Enhancement - The default root partition size has been increased to 6 GB (up from 4 GB).

Enhancement - SGE ganglia monitor added. The state of all SGE jobs can be tracked from the frontend's web page.

Enhancement - PXE support extended to support floppy-based Etherboot and ia64.

Enhancement - EKV uses ssh instead of telnet for security.

Enhancement - New Myrinet MPICH version 1.2.5..12.

Enhancement, Java Roll -- Updated JDK to version 1.4.2_04

Enhancement - Latest software updates recompiled for three architectures from RHEL source rpms.

Enhancement - Automatic MySQL Cluster database backup.

Enhancement - MAC addresses are included for each node in the "Cluster Labels" output.

Enhancement - Frontend rescue mode on the Rocks Base CD enabled. By typing "frontend rescue" at the boot prompt will give you a shell in which you can examine the state of the frontend.

Bug Fix - 411 hardened. More reliable notification of changed files. Correct Makefile encrypts login files on frontend first-boot.

Bug Fix - Multiple CD drives are supported for bringing up a frontend. If you have more than one CD drive connected to your frontend, the installer will now correctly identify which CD you are using.

Bug Fix - Ganglia metrics are now saved on frontend reboot. After a reboot, all Ganglia history will be restored from the previous boot.

Bug Fix - PVFS compiled with -mcmodel=kernel on Opteron.

Bug Fix - XML escape characters (e.g., &, <, >) are supported in the installation screens (e.g., the Cluster Information screen and the Root Password screen).

Bug Fix, Intel Roll - All the Intel compiler libraries are now copied to the compute nodes.

B.8. Release 3.2.0 - changes from 3.1.0

New Feature - Added the Condor Roll. This brings the distributed high-throughput features from the Condor project to Rocks clusters.

New Feature - Added the Area51 Roll. This roll contains security tools and services to check the integrity of the files and operating system on your cluster.

New Feature - Ganglia RSS news event service.

Enhancement - Improved network handling for compute nodes: any interface may be used for the cluster private network, not simply the default "eth0".

Enhancement - Better support for cross-architecture clusters containing x86 and x86_64 machines.

Enhancement - GM device driver now builds and loads on compute nodes that have a custom kernel (e.g., a kernel from kernel.org).

Enhancement - Software RAID for custom compute node partitioning is supported.

Enhancement - Added variables for root and swap partition. If you only want to change the size of root and/or swap, you only have to reassign two XML variables.

Enhancement - The default root partition size has been increased to 6 GB (up from 4 GB).

Enhancement - SGE ganglia monitor added. The state of all SGE jobs can be tracked from the frontend's web page.

Enhancement - PXE support extended to support floppy-based Etherboot and ia64.

Enhancement - EKV uses ssh instead of telnet for security.

Enhancement - New Myrinet MPICH version 1.2.5..12.

Enhancement, Java Roll -- Updated JDK to version 1.4.2_04

Enhancement - Latest software updates recompiled for three architectures from RHEL source rpms.

Enhancement - Automatic MySQL Cluster database backup.

Enhancement - MAC addresses are included for each node in the "Cluster Labels" output.

Enhancement - Frontend rescue mode on the Rocks Base CD enabled. By typing "frontend rescue" at the boot prompt will give you a shell in which you can examine the state of the frontend.

Bug Fix - 411 hardened. More reliable notification of changed files. Correct Makefile encrypts login files on frontend first-boot.

Bug Fix - Multiple CD drives are supported for bringing up a frontend. If you have more than one CD drive connected to your frontend, the installer will now correctly identify which CD you are using.

Bug Fix - Ganglia metrics are now saved on frontend reboot. After a reboot, all Ganglia history will be restored from the previous boot.

Bug Fix - PVFS compiled with -mcmodel=kernel on Opteron.

Bug Fix - XML escape characters (e.g., &, <, >) are supported in the installation screens (e.g., the Cluster Information screen and the Root Password screen).

Bug Fix, Intel Roll - All the Intel compiler libraries are now copied to the compute nodes.

B.9. Release 3.1.0 - changes from 3.0.0

Base Linux packages compiled from publicly available RedHat Enterprise Linux 3 Source (Advanced Workstation) for all architectures.

Switched to Sun Grid Engine 5.3 as the default batch scheduling system.

More Rolls: NMI/Globus Release 4, Java, Condor, Intel compiler rolls available.

New Architectures: Opteron (x86_64) receives first-class functionality.

Enhancement - New MPICH version 1.2.5.2. More efficient MPD parallel job-launcher handling. MPICH2 included by default as well.

Enhancement - Using latest Myrinet mpich-gm 2.0.8 for all architectures.

Enhancement - Updated SSH version 3.7.1 with no login delay.

Enhancement - 411 Secure Information Service used by default, replacing NIS.

Enhancement - Greceptor replaces Gschedule to support mpdring, 411, cluster-top and others. Achieves an order of magnitude better performance than its predecessor.

B.10. Release 3.0.0 - changes from 2.3.2

Based on RedHat 7.3 for x86 and RedHat Advanced Workstation 2.1 for ia64 (all packages recompiled from publicly available source).

Enhancement - Includes RedHat updated RPMS (and recompiled SRPMs for ia64), as of September 3 2003.

Enhancement - Includes kernel version 2.4.20-20.7 for x86 and version 2.4.18e.37 for ia64. Installation environment includes all drivers from the above kernel packages.

Enhancement - New full-featured DNS server and structured ".local" naming conventions within cluster.

Enhancement - Linpack (`xhpl`) works out of the box for Pentium IV and Athlon.

Enhancement - Added remove node feature to `insert-ethers`.

Enhancement - New layout of all MPICH transports. See `/opt/mpich` on the frontend for the new directory structure.

Enhancement - Add support for 'Rolls'. An x86 Rocks frontend install now requires two CDs: the Rocks Base CD and the HPC Roll. An ia64 frontend still requires only one DVD.

Enhancement - Added 'Grid' Roll. This roll includes all packages from NMI R3.1, which includes Globus, the Simple Certificate Authority, and other packages.

Enhancement - High-Performance, fault-tolerant MPD job launcher made available. Automatic MPD ring creation and healing via KAgreement-mpd protocol. (Currently in beta phase for this release)

Enhancement - New 411 Secure Information Service to replace NIS. (Currently in beta phase for this release)

Enhancement - Latest Ganglia version 2.5.4 including better webfrontend speed and streamlined appearance, and more efficient network and disk metric handling.

Enhancement - New PhpSysInfo page on compute nodes, available along with /proc link on Ganglia host view page.

Enhancement - Ganglia command line tool has new --clustersize and --alive=host options.

Enhancement - Kickstart graph now viewable from frontend web page.

Enhancement - For kickstart graph files, new <file> tags made available, with owner="root.root" and perms="ga+r" attributes. Beta phase of RCS-based tracking of all config file changes made for post-section repeatability.

Enhancement - Kickstart graph ordering is explicit. Previously the evaluation order of individual nodes depended on graph weights. Node dependencies can now be explicitly specified using <order> tags in the graph files.

Bug Fix - UNIX manual pages correctly shown (we extend /etc/man.conf)

Bug Fix - NTP now synchronizes all compute node clocks with the frontend.

Bug Fix - `add-extra-nic` now supports multiple NICs per compute node.

Bug Fix - Ganglia RRD metric histories are archived on physical disk and restored on startup.

Bug Fix - Includes NCSA's OpenPBS scalability patches. Can now launch PBS jobs that require more than 64 processors.

Bug Fix - USB keyboard works on all ia64 Tiger boxes

B.11. Release 2.3.2 - changes from 2.3.1

Bug fix - Memory leaks in the broadcastSSH gmetric python module are fixed.

Bug fix - Gmetad will not crash when long ganglia metric names are introduced in the cluster.

Bug Fix - Building MPICH-GM package correctly for AMD Athlon processors.

Bug Fix - Added PBS directories: `/opt/OpenPBS/sched_priv`, `/opt/OpenPBS/sched_logs`, `/opt/OpenPBS/undelivered`.

Bug Fix - Added `userdel` that correctly updates the NIS database.

Enhancement - The Rocks-specific Ganglia metrics are much more efficient with a new Python C extension module that publishes ganglia metrics. The PBS job-queue monitor particularly benefits from this new module.

Enhancement - Updated `rocks-boot` package to contain all the modules from the latest kernel-BOOT package.

Enhancement - The Ganglia monitor-core and webfrontend packages have been updated to the latest version 2.5.3.

Enhancement - The frontend is now a fully configured Rocks cluster build host. By checking out all the Rocks source code on a 2.3.2 frontend, one can build all the source code simply by executing `make rpm` in the directory `.../rocks/src/`.

Enhancement - Updated SGE packages from v5.3p2-4 to v5.3p3-1.

Enhancement - Added Rocks version number to `/home/install/contrib` directory structure.

B.12. Release 2.3.1 - changes from 2.3

Bug fix - Now all the installation device drivers from Red Hat's device disks are included (e.g., Broadcom's Ethernet adapters). In Rocks 2.3, only the device drivers found on Red Hat's installation boot floppy were included.

Bug fix - User-specified NIS domains are now supported (in Rocks 2.3, only 'rocks' NIS domain was supported).

Bug fix - User-specified compute node disk partitioning is now supported.

Bug fix - Sun Grid Engine commd port errors during post installation and Sun Grid Engine warnings during `insert-ethers` were fixed.

Bug fix - Building for Pentium II/III and Athlon added to ATLAS RPM. (on a side note, ATLAS is now built against gcc version 3.2).

Enhancement - PVFS upgraded to version 1.5.6.

Enhancement - More detail has been added to the PBS queue monitoring web page (e.g., can view jobs for only one user and can view nodes for one job). Additionally, the monitoring code now more efficent and it has been hardened due to direct experiences on a 300-node Rocks cluster.

Enhancement - The `bssh` service has been moved from a standalone service to a task managed by the Ganglia `gschedule` service.

Enhancement - The ethernet-based MPICH package has been updated to version 1.2.5.

Enhancement - The Myrinet-based MPICH package has been updated to version 1.2.5..9.

Enhancement - `OpenPBS` version 2.3.16 has replaced PBS. Additionally, the *big memory* patch has been applied. Also, the license for OpenPBS requires registration for those that use OpenPBS, so if you use OpenPBS to manage your computational resources, please register at http://www.OpenPBS.org.

Enhancement - The `maui` package has been updated to version 3.2.5.

Enhancement - Updated Myricom's GM to version 1.6.3.

New Feature - Added a link of the main web page of the frontend that allows one to make sheets of labels with the names of all the compute nodes.

New Feature - An alternative version of `gcc` is now installed (version 3.2 is installed in /opt/gcc32/...).

B.13. Release 2.2.1 - changes from 2.2

Bug fix - pvfs and gm modules don't build because the kernel source and kernel binary RPMs were of a different version.

Bug fix - the partitioning on compute nodes only partitioned the first drive. Now all drives on compute nodes are partitioned with a single partition. The default partitioning is: 4 GB root partition, then /state/partition1 is the remainder of the first drive. The second drive, if present, will have one partition labeled "/state/partition2". The third drive, if present, will have one partition labeled "/state/partition3", etc.

Bug fix - the Rocks CD didn't support as many hardware devices as the RedHat CD. All the hardware modules found on the RedHat CD have been added to the Rocks CD (including many, many more).

B.14. Release 2.2 - changes from 2.1.2

Based on RedHat 7.2.

Upgraded Ganglia (provided by Matt Massie of UC Berkeley) to 2.1.1.

Incorporated PVFS RPMs that were graciously provided to us from Najib Ninaba and Laurence Liew who work at Scalable Systems Pte Ltd in Singapore.

insert-ethers looks to see if a Rocks distribution exists. If it doesn't, insert-ethers rebuilds it.

Upgraded MPICH-GM to version 1.2.1..7b.

Added the "stream" memory bandwidth benchmark.

Added functionality to rocks-dist so distributions can be rebuilt without having to mirror the entire distribution.

Implemented a "greedy" partitioning scheme on compute nodes. The default partitioning is: 4 GB root partition, then /state/partition1 is the remainder of the first drive. The second drive, if present, will have one partition labeled "/state/partition2". The third drive, if present, will have one partition labeled "/state/partition3", etc.

Bug fix - added a "watchdog" timer to kickstart. This reboots a kickstarting node if it can't find a kickstart file. This problem was reported by folks trying to kickstart multiple nodes at the same time.

Bug fix - increased the polling intervals for maui so it won't time out when asking PBS about node status on larger clusters.

Bug fix - makedhcp now adds the full pathname to pxelinux.0 when it builds dhcpd.conf.

Bug fix - create a device node for /dev/cdrom.

Bug fix - /var/log/messages is now appropriately rotated.

B.15. Release 2.1.2 - changes from 2.1.1

Many network and storage drivers have been added to the installation CD. For example, SMC 83c170 EPIC/100 (epic100.o), RTL8139 SMC EZ Card Fast Ethernet (8139too.o) and the Promise SuperTrak Driver (pti_st.o) have all been included (as well as about 100 more).

The cluster configuration web form has been simplified.

The initial kickstart file that is generated from the web form is now streamed directly back to the user (rather than displaying the kickstart file, and then asking the user to save the file). This should finally kill the "I saved my kickstart file on Windows" problem.

An option to manually partition a frontend disk has been added to the cluster configuration web form.

The recursive directory /home/install/install/install/... has been eliminated.

Ganglia's axon is now started before pbs-server, as the pbs-server initialization script asks ganglia for the number of processor in each node when it creates one of it's configuration files.

The latest "stable" release of Myricom's GM (1.5) and MPICH-GM (1.2.1..7) packages.

High-Performance Linpack is now precompiled for Myrinet and Ethernet.

B.16. Release 2.1.1 - changes from 2.1

The main change in this release is the use of an XML-based *kickstart graph* to actively manage kickstart files.

Includes support for IA-64 compute nodes. See the Installing IA-64 Compute Nodes HOWTO[4] for detailed information.

A full X server is now installed on frontend machines.

Added PXE support for kickstarting compute nodes.

All compute nodes now install ATLAS and high-performance Linpack -- some slick software from the Innovative Computing Laboratory[5] at the University of Tennessee.

Modified to the PBS server initialization script to dynamically determine the number of CPUs in compute nodes by querying `ganglia`.

Created a `rocks-pylib` package that contains all the common code used by Rocks command line utilties that access the MySQL database, thus giving all the tools the same basic functionality and common user-specified flags.

Patched Red Hat's installation tool (anaconda) so the default behavior is to get kickstart files with HTTP (Red Hat's default is NFS). This frees the installation procedure of requiring NFS for *any* of its functions.

Rewrite of `insert-ethers` to give it the look and feel of a standard Red Hat installation tool.

Now using Red Hat's `pump` instead of `dhclient` for the DHCP client.

Properly create the default PBS configuration file (`/usr/apps/pbs/pbs.default`) so PBS is now operational "out of the box".

Fixed the annoying, but harmless, message "`socket.error: (101, 'Network is unreachable')`" that was seen on frontend boots.

Fixed the annoying, but harmless, message "`user 0 unknown`" that was seen on a compute node's first boot after kickstarting.

Fixed the 444 permissions problem on `/usr/man` and moved all the Rocks man pages into the new home for Linux man pages (`/usr/share/man`).

B.17. Release 2.1 - changes from 2.0.1

The main change in this release is that thanks to RedHat 7.1, we now use the Linux 2.4 kernel.

Based on RedHat 7.1, instead of 7.0.

Linux 2.4.x kernel, instead of 2.2.x.

Cluster-dist has been replaced with Rocks-dist. Command line arguments are very similar, with the `explode` command being removed and replaced with the `--copy` flag. The new Rocks-dist creates smaller distributions, fixes the problem of expensive mirror updating, and simplifies CD building. Also, it no longer deletes the distribution before rebuilding, this means the build directory (where kickstart files reside) is persistent across distribution builds.

Frontend is now a stratum 10 NTP server, so compute nodes will clock sync to the frontend even when the frontend cannot reach an external time source.

Usher daemon now correctly daemonizes, since we patch the GM code to allow processes to fork.

Symbolic links for Ekv and piece-pipe RPMs removed from the build directory, and "`@Control@`" section added to kickstart files.

Pbs_mom_config.h generated in the kickstart build directory.

Added pre-defined types to the models table in the SQL database. Also, removed dead tables from database, and made column order more human friendly.

Add SQL parsing to cluster-[ps|kill|fork] scripts.

Removed cluster-config-compute, and cluster-config-frontend from the "%post" section in the kickstart file. The cluster-config rpm is now build and installed on the fly on each compute-node.

Bumped lilo timeout to 5 seconds.

Added FORCE_UNIPROCESSOR macro test to force sick SMP machines to kickstart as uniprocessor nodes.

Major revision of insert-ethers. Can now be used to replace nodes, and start at arbitrary ranks and basenames.

Minor maui and pbs bug fixes.

Added gm-mpich SHMEM support to mpi-launch.

B.18. Release 2.0.1 - changes from 2.0

Changed to new directory structure according to RedHat. Existing users will have to delete their mirror of www.rocksclusters.org and re-mirror to pickup the current RedHat directory naming scheme. NOTE: you need the new cluster-dist from www.rocksclusters.org to create a new mirror!

Added support to kickstart laptops (still working on this)

Frontend can now have either a DHCP or static address for the external network. For DHCP the DNS information provided from the external DHCP server is inserted into the Rocks Database and propagated to compute nodes.

Increased default DHCP lease time

Replaced Linux's useradd with create-account.

Force glibc-common RPM to be installed. RedHat 7.0 doesn't install this due to errors in the RPM database.

NIS database gets rebuilt on the frontend once an hour.

Create directories on frontend/compute nodes before putting down SSL and SSH keys. Fixed permission on directories.

Ssh-agent now forwards through nodes

Ssh doesn't use privileged port (makes firewalls happy)

cluster-kickstart set real and effect UID to root so all members of the install group can run shoot-node. Previously only root could do this.

Fixed reinstalls on IDE and SCSI hosts (only IDA host worked before, thanks to a RedHat 7.0 change)

Fixed bssh bug

Notes

1. http://www.rocksclusters.org/roll-documentation/base/4.3/commandline.html
2. http://www.rocksclusters.org/roll-documentation/base/4.3/c229.html
3. http://www.rocksclusters.org/roll-documentation/base/4.3/boot-order.html
4. ../howto/ia64.php
5. http://icl.cs.utk.edu/

Appendix C. 411 Secure Information Service Internals

The 411 Secure Information Service provides NIS-like functionality for Rocks clusters. It is named after the common "411" code for information in the phone system. We use 411 to securely distribute password files, user and group configuration files and the like.

411 uses Public Key Cryptography to verify file contents, and shared key cryptography to protect transport. It operates on a file level, rather than the RPC-based per-line maps of NIS. 411 does not rely on RPC, and instead distributes the files themselves using HTTP (web service). Its central task is to securely maintain critical login/password files on the worker nodes of a cluster. It does this by implementing a file-based distributed database with weak consistency semantics. The design goals of 411 include scalability, security, low-latency when changes occur, and resilience to failures.

C.1. Using the 411 Service

The 411 system intentionally mimics the NIS interface for system administrators. Of course there are elements in 411 which are not present in NIS, namely RSA cryptographic keys. However we have attempted to make 411 as easy to use as an NIS replacement.

Files listed in `/var/411/Files.mk` are automatically serviced by 411. This means that any file listed there will be kept up to date by the 411 agents on all compute nodes in your cluster. This is done using the makefile `/var/411/Makefile` in a similar fashion to NIS. To force the 411 system to flush all changes, execute the following on the frontend node:

```
# make -C /var/411
```

Note that this command is run by cron every hour on the frontend to propagate changes to compute nodes. New files can be added to Files.mk as necessary for custom services on the cluster.

To force all 411 files to be re-encrypted and change alerts sent to all compute nodes, run this on the frontend

```
# make -C /var/411 force
```

The 411 service uses IP broadcast messages on your cluster's private network to achieve optimal performance.

To force all compute nodes to retrieve the latest files from the frontend, execute:

```
# rocks run host command="411get --all"
```

C.2. Structure

C.2.1. Listener

The 411 system uses a Rocks RPC service called channeld to communicate with client nodes. The client nodes run the RPC service, and the frontend has a RPC client which sends out alerts on its multicast channel. When the clients receive a 411-alert RPC message, along with the name of the file that changed, it does a pull of the file using 411get.

411 is akin to a distributed database, and is not a centralized lookup service like NIS. While scalable, 411 does not provide instantaneous distribution of new files. The delay between running the 411 makefile and all nodes receiving the changed file depends on cluster size. A large password file on a cluster with many nodes can take up to a minute to fully synchronize on all nodes.

C.3. 411 Groups

Beginning in Rocks 3.3.0, 411 has the ability to send messages to subsets of the cluster. This facility, called 411 groups, allows us to distribute different files to nodes depending on their type. The group mechanism depends on the client nodes specifying group names in their local 411 configuration file; these are called the client's "registered" groups.

There is no per-group key in 411. The groups mechanism is only a convenience feature, without strong security to enforce it. Specifically, a node can eavesdrop on messages for a foreign group that it is not a member of.

Group names are multi-level, and resemble file paths. By default, every node is a member of the '/' group (corresponding to the traditional top-level 411 group), and the '/Membership' group, where *membership* is the node membership in the frontend database, such as "Compute" or "NAS".

A special Makefile called /var/411/Group.mk is available to help you setup and maintain 411 groups. After editing this file to specify which files go to which group, run

```
# make -C /var/411 groups
# make -C /var/411
```

To activate the 411 group makefile actions.

By default, nodes are members of a group with the same name as their *Membership*. For example compute nodes are automatically a member of the group "Compute". A sample `411.conf` file with several groups looks like:

```
<!-\- Configuration file for the 411 Information Service -\->
<config>
<master url="http://10.1.1.1:372/411.d/" score="0"/>
<group>/light/blue</group>
<group>Compute</group>
</config>
```

Multi-element group names have a simple inheritance model: specific groups imply more general ones. For example, if you are a member of the group `/compute/light`, you will automatically be interested in messages in group "/compute/light" and "/compute". You will not be interested in messages from group "/compute/heavy". In this case "/compute/light" is the specific group, and "/compute" is the more general one.

C.4. Plugins

Starting with Rocks 5.4, file specific plugins have been introduced into the 411 system. These plugins manipulate the content of files before sending files on the frontend, and after reception of the file on the client nodes. The plugins also mangle name of the file, ownership and mode of the file.

C.4.1. Plugin API

The 411 Plugin architecture follows a very simple API. Each plugin is written in Python and the plugins reside in `/opt/rocks/var/plugins/411/`. All plugins inherit the `rocks.service411.Plugin` class.

1. `get_filename`:This function returns a the filename on the frontend on which this plugin will function. This is the only required function in the plugin. All other functions are optional.

   ```
   import os
   import sys
   import stat
   import rocks.service411

   class Plugin(rocks.service411.Plugin):
    def get_filename(self):
      return '/etc/auto.master'
   ```

2. `filter_name`: This function mangles the filename of the file at the destination, and returns the mangled name

   ```
   def filter_name(self, fname):
    if fname == '/etc/auto.master' && self.attrs['os'] == 'sunos':
     return '/etc/auto_master'
   ```

3. `filter_owner`: This function takes the Owner UID and group ID of the file as a string argument and returns a mangled version of the "UID.GID" string. This string is in the format that the **chown** command understands.

   ```
   def filter_owner(self, oid):
    if self.attrs['os'] == 'linux':
     return oid
    if self.attrs['os'] == 'sunos':
     return '0.0'
   ```

4. `filter_mode`: This function takes the mode information as a string, and returns a mangled mode information string. This should be in the numerical format that the **chmod** command understands.

   ```
   def filter_mode(self, mode):
    return '010644'
   ```

5. `filter_content`: This function takes the contents of the file as a string, manipulates it and returns the final string to be stored in the file. The example below illustrates insertion of a blank line between every line of the input content.

   Example C-1. filter_content

   ```
   def filter_content(self, content):
    new_content = []
    for line in content.split('\n'):
     new_content.append(line + '\n' + '\n')
    return ''.join(new_content).strip()
   ```

6. `pre_send`: This function manipulates the contents of the file before the file is made available for download over 411. The example below illustrates how deletion of all blank lines from the content.

   Example C-2. pre_send

   ```
   def pre_send(self, content):
    new_content = []
    for line in content.split('\n'):
     if line.strip() is '':
      continue
     else:
      new_content.append(line)
    return '\n'.join(new_content).strip()
   ```

C.5. 411get Configuration File

As mentioned above, 411get configuration is stored in a file on the client nodes. This file is generated by a rocks command.

```
#rocks report host config411 compute-0-0
<file name="/etc/411.conf" perms="0600" owner="root:root">
<![CDATA[<!-- 411 Configuration -->
<config>
<appliance>compute</appliance>
<master url="http://10.1.1.1:372/411.d/"/>
<os>linux</os>
```

```
<group>Compute</group>
</config>]]>
</file>
```

The configuration file contains information about the client node, stored as XML entities. For eg. - the appliance, group, and OS information. This entities are converted to attributes that are made available to the plugins. When 411get is run, the config file is read, and the OS, appliance, etc. information is stored in a dictionary called `self.attr`. This dictionary is made available to all the plugins, so that the plugins may filter based on the attribute of the nodes.

The admin may extend the config file to include entities other than the default ones. For this a rocks command line plugin must be created that outputs XML tags. Please see how to create a rocks command line plugin in the Rocks Developers' Guide.

Once the rocks command line plugin is created for the **rocks report host config411** command, the configuration can be synced to the client nodes using the following commands.

```
# rocks report host config411 compute-0-0  | ssh compute-0-0 "rocks report script | sh"
```

C.6. Commands

C.6.1. 411get

```
411get [--all] [--master=url] [--conf] [--pub] [--shared] [--local] [file]
```

Retrieves and decrypts 411 messages. Prints resulting file to stdout. When invoked with no files, 411get will list the available 411 messages.

The following options are available:

- *--all* Retrieves and writes all available 411 messages from the most attractive master. Does not print output to stdout, nor ask for confirmation before overwriting files.
- *--master* The url of a 411 master server to use. Defaults to "http://10.1.1.1:372/411.d/" or whatever is present in "/etc/411.conf". If given, this master takes precedence over those listed in the configuration file.
- *--file, --local* Assume the file is local, ie present in the current directory. Does not use http to retrieve the file. Decrypts and prints the file contents.
- *--conf* The configuration file to use. Defaults to "/etc/411.conf".
- *--pub* The location of the cluster public RSA key. Defaults to `/etc/411-security/master.pub`.
- *--shared* The location of the cluster shared key. Defaults to `/etc/411-security/shared.key`

The master servers, along with their quality score, are listed in the `/etc/411.conf` file on compute nodes.

C.6.2. 411put

```
411put [--411dir=dir] [--urldir=dir] [--see] [--noalert] [--alert=channel]
[--411name] [--pub] [--priv] [--comment=char] [--chroot=dir]
[--chroot-here] [--group=group] file1 file2 ...
```

Encrypts and publishes files using the 411 secure information service. Will send a broadcast message to client nodes by default, alerting them of a changed file.

The following options are available:

- *--chroot=dir* Turn "dir" into the root directory of the destination file. This allows files to be located in a different place on the master and clients.

  ```
  Example:
  411put --chroot=/var/411/groups/compute /var/411/groups/compute/etc/passwd
  ```

  Will put "/var/411/groups/compute/etc/passwd" on compute nodes as "/etc/passwd".
- *--chroot-here* A convenience option, equivalent to --chroot=$PWD.
- *--group=name* A 411 group for this file. Clients will ignore 411 messages in groups which they are not a part of. Allows 411 files to be published to a subset of the cluster. Name is path-like: "Compute/green", or "/Compute/green". Spaces are ok: "a space/yellow" is a valid group name as well.
- *--comment* The comment character for this file. Used to place a descriptive header without disrupting normal operations. Often set to "#". Default is none.
- *--411dir* The local directory to place encrypted 411 messages. Defaults to "/etc/411.d/". Be careful about the permissions of this directory.
- *--urldir* The web directory where 411 messages are available. Defaults to "/411.d/".
- *--see* Shows the encrypted file contents on stdout.
- *--noalert* Suppresses alert message.
- *--411name* Prints the 411 message name for the file. Provided for convenience.
- *--pub* The location of the 411 master public RSA key. Defaults to a 1024 bit key in "/etc/411-security/master.pub". This file should have permissions 0444 (read by all) and be owned by root.
- *--priv* The location of the 411 master private RSA key. Defaults to a 1024 bit key in "/etc/411-security/master.key". This file should exist only on the master node and be owned by root and have permissions 0400 (read only by root).
- *--make-shared-key* Generate a new random shared key. The key is a 256 random number encoded in base64.

Appendix D. Changes to Rocks Security Infrastructure

D.1. Rocks Password Infrastructure

D.1.1. Current Design

Currently, the root password for the frontend is elicited from the user during the frontend installation. Once the password is obtained, 3 crypted versions are generated from the root password. One version is the UNIX crypt version, which forms the root password for the cluster, another is the portable password format which is used for wordpress access. The third is an SHA version which is used as the MySQL root password. These three crypted passwords are stored in the database as shadow attributes.

When compute nodes are installed the root password that is stored in the database is transferred over to the compute node as an XML entity. This way we maintain root password consistency between client nodes and frontend. Any changes made to the root password on the frontend using the **passwd** command is propogated to the compute nodes using the 411 system.

D.1.2. Changes to the Design

From the above description of the existing design, we can see that the root password is stored in multiple places and used for multiple purposes. We determined that this was not the optimal security situation, and could use changes.

The way we've implemented changes are as follows:

1. As soon as the root password is obtained from the admin, it is converted to only the UNIX crypt format. It is used during the installation to set the root password, and is not used for any other purposes, and is not stored in the database.
2. Random passwords are generated for root access to the database and access to the wordpress engine.
3. The root account on each compute node gets a random root password.
4. The 411 system uses plugins to filter the content of the password and shadow files so that it does not transmit information about any accounts with UID under 500. Therefore, any changes made to the root password on the frontend is not propagated to the compute nodes.
5. To change the root password for the compute nodes, the admin may now use the "secure attribute" infrastructure in Rocks. The admin may also use the **passwd** command on the compute nodes to change the password without fear of override.

   You may NOT use both the secure attributes and the passwd command to change the password on the compute nodes. If you do, the value in the secure attributes table always wins.

6. To change the root password on the frontend, you can now use the **passwd**command, and not worry about changing any of the other passwords in the cluster.

D.1.3. Changing the root password on a compute node

This section describes how to change the root password for compute nodes. It uses the rocks secure attribute infrastructure

```
# rocks set host sec_attr compute attr=root_pw
  Enter root_pw:
Confirm root_pw:
# rocks list host sec_attr compute
HOST         ATTR    VALUE                              ENC
compute-0-0: root_pw $1$G55uZRrO$5MI7Nv14U5yWfKAlauqPE0 crypt
compute-0-1: root_pw $1$G55uZRrO$5MI7Nv14U5yWfKAlauqPE0 crypt
# rocks sync host sec_attr compute
```

1. **rocks set host sec_attr** creates the entry in the database.
2. **rocks sync host sec_attr** changes the root password on the compute nodes.

The admin can also set different root password for each node

D.2. Rocks Secure Attribute Infrastructure

As a replacement for shadow attributes, we introduced the Rocks secure attributes infrastructure. The reason for the change was internal, as shadow attributes were still accessible by the apache user. Secure attributes are not accessible by anyone but the root user.

The secure attributes are not synced along with any of the other regular attributes. The secure attributes are synced between hosts completely out of band using standard SSH.

D.2.1. Structure

The secure attributes infrastructure consists of two parts.

1. The secure attribute itself, which is stored in the database.
2. Plugins, in the form of python code, which defines how to act on the secure attribute. These plugins reside in `/opt/rocks/var/plugins/sec_attr/`. The plugins have a very simple API.

D.2.2. Plugins API

The Rocks secure attribute plugin class inherits a single class called `rocks.commands.sec_attr_plugin`.

This example is a very simple fictitious use-case. A service called **fake_service** runs on all compute nodes, and stores an SHA password in it's configuration file. The configuration file for this service is called `/etc/fake_service.conf`. These are the contents of the config file.

```
# cat /etc/fake_service.conf
password=150b95f90c06f127a040a40f98582231369b6fda
```

D.2.2.1. Setting the Secure Attribute

This password can be stored in the database as a secure attribute. To store the password as a secure attribute, run

```
# rocks set host sec_attr compute attr=fake_svc_pw enc=sha
  Enter fake_svc_pw:
Confirm fake_svc_pw:
```

Enter the password string and confirm it (the same way you would when changing the root password).

D.2.2.2. Creating the plugin

Secure attribute plugins are located in the `/opt/rocks/var/plugins/sec_attr` directory. These plugins are written in python, and contain very simple API. There are two functions that must be a part of all plugins.

1. `get_sec_attr`: This function returns a the name of the secure attribute as stored in the database. In the example, this would return `fake_svc_pw`.

   This name should match exactly the attribute name in the database. If multiple plugins return the same value for this function, then the last match wins.

2. `filter`: This function is the workhorse of the plugin. This takes in a single value , and performs the desired manipulation, filtering, and storage on it.

The plugin for the above example is `/opt/rocks/var/plugins/sec_attr/fake.py`. The name of the file is irrelevant. The command iterates over all available plugins and uses the last match on the return value of `get_sec_attr` function.

Example D-1. Plugin code for fake_svc_pw - fake.py

```
import rocks.commands
import os, sys, string

class plugin(rocks.commands.sec_attr_plugin):
```

```
 def get_sec_attr(self):
  return 'fake_svc_pw'

 def filter(self, value):
  f = open('/etc/fake_service.conf','w')
  f.write('password=%s\n' % value)
  f.close()
```

D.2.2.3. Syncing the attribute

Once the plugin has been created and the attribute has been assigned in the database, you can sync the plugin using the command

```
# rocks sync host sec_attr
# rocks run host compute command="cat /etc/fake_service.conf" collate=yes
compute-0-1: password=150b95f90c06f127a040a40f98582231369b6fda
compute-0-0: password=150b95f90c06f127a040a40f98582231369b6fda
```

Appendix E. Kickstart Nodes Reference

E.1. Rocks Base Nodes

E.1.1. 411

The packages and other common elements of the 411 Secure Information Service.

Parent Nodes:

- base

E.1.2. 411-client

Sets up the 411 Secure Information Service for clients. The 411 service will automatically configure itself when a file is published. Also puts all current 411 files from the frontend into the kickstart file for services that cannot tolerate a single 411 failure. Note that 411 can never guarentee full absolute success at any single time. It only offers consistancy over the long term.

Parent Nodes:

- client

E.1.3. 411-server

Sets up the 411 Secure Information Service for Master nodes. Creates the RSA public and private keys for the cluster, and configures Apache for 411.

Parent Nodes:

- server

E.1.4. apache

Apache HTTP Server

Parent Nodes:

- base

- cluster-db

E.1.5. autofs

AutoFS for automounting home directories over NFS or the loopback device.

Parent Nodes:

- autofs-client
- autofs-server

E.1.6. autofs-client

AutoFS Client

Parent Nodes:

- client

Children Nodes:

- autofs

E.1.7. autofs-server

AutoFS server

Parent Nodes:

- server

Children Nodes:

- autofs

E.1.8. base

Base class for all Rocks nodes. This should include compute nodes, frontend nodes, standalone laptops, computer labs, graphics nodes, nfs servers To achieve this level of flexibility this base class should have edges only to those classes that implement the core of Rocks.

Parent Nodes:

- client
- server

Children Nodes:

- 411
- apache
- c-development
- disk-stamp
- elilo
- fstab
- grub
- installclass
- ip-diag
- keyboard
- logrotate
- node
- node-thin
- rpc
- scripting
- ssh
- ssl

E.1.9. c-development

Minimalist C development support. This is everything you need to compile the kernel.

Parent Nodes:

- base

E.1.10. cdr

CDR Tools (burnings, iso, ripping, mp3 enconding)

Parent Nodes:

- devel

E.1.11. central

A Rocks Cluster Central server. Can kickstart other servers over the network.

Parent Nodes:

- server

E.1.12. client

The 'client node' in the graph. This file is used as a connection point for other XML configuration nodes.

Children Nodes:

- 411-client
- autofs-client
- base
- installclass-client
- ntp-client
- ssh-client
- syslog-client

E.1.13. cluster-db

Rocks Cluster Database

Parent Nodes:

- server

Children Nodes:

- apache

E.1.14. cluster-db-data

Populate cluster database with initial data

Parent Nodes:

- server

E.1.15. cluster-db-structure

Cluster Database SQL table structure. This used to be generated from a dump of the structure on Meteor. Now we just edit this directly.

Parent Nodes:

- server

E.1.16. devel

The ’devel node’ in the graph. This file is used as a connection point for other XML configuration nodes.

Parent Nodes:

- server

Children Nodes:

- cdr
- docbook
- emacs
- fortran-development

E.1.17. dhcp-server

Setup the DHCP server for the cluster

Parent Nodes:

- server

E.1.18. disk-stamp

Take a root partition, and make it ours! This is the key to determining, on reinstalls, if we should save partitions (because the stamp is there) or blow away all the partitions on the disk (because the stamp isn't there).

Parent Nodes:

- base

E.1.19. dns-server

Configures a DNS nameserver for the cluster on the frontend. Both forward and reversed zones are defined using the database.

Parent Nodes:

- server

E.1.20. docbook

DOC Book support (needed to build rolls)

Parent Nodes:

- devel

E.1.21. elilo

IA-64 Bootloader support

Parent Nodes:

- base

E.1.22. emacs

Emacs OS

Parent Nodes:

- devel

E.1.23. fortran-development

Fortran

Parent Nodes:

- devel

E.1.24. fstab

Examine the disks on the box we're installing and see if there are existing, non-root partitions which we should preserve.

Parent Nodes:

- base

E.1.25. grub

IA-32 Boot loader support

Parent Nodes:

- base

E.1.26. install

Do everything needed to kickstart compute nodes or, generally speaking, everything needed to kickstart any node from this machine.

Parent Nodes:

- server

E.1.27. installclass

The base installclass files. This graph node must preceed any other installclass graph nodes.

Parent Nodes:

- base

E.1.28. installclass-client

The client installclass files.

Parent Nodes:

- client

E.1.29. installclass-server

The server installclass files.

Parent Nodes:

- server

E.1.30. ip-diag

TCP/IP Network diagnostic tools.

Parent Nodes:

- base

E.1.31. keyboard

Support USB keyboard for ia64

Parent Nodes:

- base

E.1.32. logrotate

Append rules to logrotate to prune files in /var/log

Parent Nodes:

- base

E.1.33. media-server

Root for the kickstart file on the CD/DVD.

Children Nodes:

- server

E.1.34. node

A node is a machine in the cluster. Node's are on a private network and get DHCP/NIS state from the frontend.

Parent Nodes:

- base

E.1.35. node-thin

Turn off a bunch of packages we think we can live without. They take up too much room on the CD. For DVD based systems this is not required Be the ugly american. the only reason why we do this is because we want to be able to fit a rocks-enabled solution onto a single cdrom and the packages below don't directly help people to run parallel applications

Parent Nodes:

- base

E.1.36. ntp

Network Time Protocol

Parent Nodes:

- ntp-client
- ntp-server

E.1.37. ntp-client

Network Time Protocl

Parent Nodes:

- client

Children Nodes:

- ntp

E.1.38. ntp-server

Network Time Protocl

Parent Nodes:

- server

Children Nodes:

- ntp

E.1.39. perl-development

Perl support

Parent Nodes:

- scripting

E.1.40. python-development

Python support

Parent Nodes:

- scripting

E.1.41. rocks-dist

Distribution building with rocks-dist

Parent Nodes:

- server

E.1.42. rpc

RPC support

Parent Nodes:

- base

E.1.43. scripting

Parent Nodes:

- base

Children Nodes:

- perl-development
- python-development
- tcl-development

E.1.44. server

The 'server node' in the graph. This file is used as a connection point for other XML configuration nodes.

Parent Nodes:

- media-server
- server-wan

Children Nodes:

- 411-server
- autofs-server
- base
- central
- cluster-db
- cluster-db-data
- cluster-db-structure
- devel
- dhcp-server
- dns-server
- install
- installclass-server
- ntp-server
- rocks-dist
- syslog-server
- x11-thin

E.1.45. server-wan

A Rocks Cluster machine that has been kickstarted over the wide area network. Used by the central server to construct a minimal kickstart file.

Children Nodes:

- server

E.1.46. ssh

Enable SSH

Parent Nodes:

- base

E.1.47. ssh-client

SSH Config for compute nodes and other non-frontend appliances. We are using one key pair among all SSH servers in the cluster. This implies we do not care about Man-in-the-Middle attacks. We have subverted the protection for these attacks for several releases (broadcastSSH). This logic should not be in the ssh.xml node so the frontend will generate its own keypair.

Parent Nodes:

- client

E.1.48. ssl

Open SSL support

Parent Nodes:

- base

E.1.49. syslog

Setup Syslog

Parent Nodes:

- syslog-client
- syslog-server

E.1.50. syslog-client

Setup Syslog for client machine to forward messages

Parent Nodes:

- client

Children Nodes:

- syslog

E.1.51. syslog-server

Setup Syslog for server to accept forwarded messages

Parent Nodes:

- server

Children Nodes:

- syslog

E.1.52. tcl-development

Tcl support

Parent Nodes:

- scripting

E.1.53. x11

X11 Desktop applications.

Parent Nodes:

- x11-thin

E.1.54. x11-thin

Trimmed down version of X11 for when we don't need sound all all that other GUI nonsense. I just want to run netscape man.

Parent Nodes:

- server

Children Nodes:

- x11

Appendix F. Rocks® Copyright and Trademark

F.1. Copyright Statement

```
                                Rocks(r)
                         www.rocksclusters.org
                         version 5.4.3 (Viper)

Copyright (c) 2000 - 2011 The Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice unmodified and in its entirety, this list of conditions and the
following disclaimer in the documentation and/or other materials provided
with the distribution.

3. All advertising and press materials, printed or electronic, mentioning
features or use of this software must display the following acknowledgement:

        "This product includes software developed by the Rocks(r)
        Development Team at the San Diego Supercomputer Center at the
        University of California, San Diego and its contributors."

4. Except as permitted for the purposes of acknowledgment in paragraph 3,
neither the name or logo of this software nor the names of its
authors may be used to endorse or promote products derived from this
software without specific prior written permission.  The name of the
software includes the following terms, and any derivatives thereof:
"Rocks", "Rocks Clusters", and "Avalanche Installer".  For licensing of
the associated name, interested parties should contact Technology
Transfer & Intellectual Property Services, University of California,
San Diego, 9500 Gilman Drive, Mail Code 0910, La Jolla, CA 92093-0910,
Ph: (858) 534-5815, FAX: (858) 534-7345, E-MAIL:invent@ucsd.edu

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS
BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
```

```
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN
IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

F.2. Trademark Licensing

The Rocks® logo (Figure F-1) and the names Rocks®, Rocks Clusters™, and Avalanche Installer™ have been submitted to the United States Patent and Trademark Office to become registered trademarks.

Commercial entities that wish to use any of the above names or logo in a product name or marketing material, are required to get written permission from the Technology Transfer and Intellectual Property Services Office at UCSD <invent@ucsd.edu>.

Figure F-1. Rocks® logo

Appendix G. Common Licenses

G.1. Artistic License

```
    The "Artistic License"

    Preamble

The intent of this document is to state the conditions under which a
Package may be copied, such that the Copyright Holder maintains some
semblance of artistic control over the development of the package,
while giving the users of the package the right to use and distribute
the Package in a more-or-less customary fashion, plus the right to make
reasonable modifications.

Definitions:

 "Package" refers to the collection of files distributed by the
 Copyright Holder, and derivatives of that collection of files
 created through textual modification.

 "Standard Version" refers to such a Package if it has not been
 modified, or has been modified in accordance with the wishes
 of the Copyright Holder as specified below.

 "Copyright Holder" is whoever is named in the copyright or
 copyrights for the package.

 "You" is you, if you're thinking about copying or distributing
 this Package.

 "Reasonable copying fee" is whatever you can justify on the
 basis of media cost, duplication charges, time of people involved,
 and so on.  (You will not be required to justify it to the
 Copyright Holder, but only to the computing community at large
 as a market that must bear the fee.)

 "Freely Available" means that no fee is charged for the item
 itself, though there may be fees involved in handling the item.
 It also means that recipients of the item may redistribute it
 under the same conditions they received it.

1. You may make and give away verbatim copies of the source form of the
Standard Version of this Package without restriction, provided that you
duplicate all of the original copyright notices and associated disclaimers.

2. You may apply bug fixes, portability fixes and other modifications
derived from the Public Domain or from the Copyright Holder.  A Package
modified in such a way shall still be considered the Standard Version.
```

```
3. You may otherwise modify your copy of this Package in any way, provided
that you insert a prominent notice in each changed file stating how and
when you changed that file, and provided that you do at least ONE of the
following:

    a) place your modifications in the Public Domain or otherwise make them
    Freely Available, such as by posting said modifications to Usenet or
    an equivalent medium, or placing the modifications on a major archive
    site such as uunet.uu.net, or by allowing the Copyright Holder to include
    your modifications in the Standard Version of the Package.

    b) use the modified Package only within your corporation or organization.

    c) rename any non-standard executables so the names do not conflict
    with standard executables, which must also be provided, and provide
    a separate manual page for each non-standard executable that clearly
    documents how it differs from the Standard Version.

    d) make other distribution arrangements with the Copyright Holder.

4. You may distribute the programs of this Package in object code or
executable form, provided that you do at least ONE of the following:

    a) distribute a Standard Version of the executables and library files,
    together with instructions (in the manual page or equivalent) on where
    to get the Standard Version.

    b) accompany the distribution with the machine-readable source of
    the Package with your modifications.

    c) give non-standard executables non-standard names, and clearly
    document the differences in manual pages (or equivalent), together
    with instructions on where to get the Standard Version.

    d) make other distribution arrangements with the Copyright Holder.

5. You may charge a reasonable copying fee for any distribution of this
Package.  You may charge any fee you choose for support of this
Package.  You may not charge a fee for this Package itself.  However,
you may distribute this Package in aggregate with other (possibly
commercial) programs as part of a larger (possibly commercial) software
distribution provided that you do not advertise this Package as a
product of your own.  You may embed this Package's interpreter within
an executable of yours (by linking); this shall be construed as a mere
form of aggregation, provided that the complete Standard Version of the
interpreter is so embedded.

6. The scripts and library files supplied as input to or produced as
output from the programs of this Package do not automatically fall
under the copyright of this Package, but belong to whoever generated
them, and may be sold commercially, and may be aggregated with this
Package.  If such scripts or library files are aggregated with this
```

```
Package via the so-called "undump" or "unexec" methods of producing a
binary executable image, then distribution of such an image shall
neither be construed as a distribution of this Package nor shall it
fall under the restrictions of Paragraphs 3 and 4, provided that you do
not represent such an executable image as a Standard Version of this
Package.

7. C subroutines (or comparably compiled subroutines in other
languages) supplied by you and linked into this Package in order to
emulate subroutines and variables of the language defined by this
Package shall not be considered part of this Package, but are the
equivalent of input as in Paragraph 6, provided these subroutines do
not change the language in any way that would cause it to fail the
regression tests for the language.

8. Aggregation of this Package with a commercial distribution is always
permitted provided that the use of this Package is embedded; that is,
when no overt attempt is made to make this Package's interfaces visible
to the end user of the commercial distribution.  Such use shall not be
construed as a distribution of this Package.

9. The name of the Copyright Holder may not be used to endorse or promote
products derived from this software without specific prior written permission.

10. THIS PACKAGE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR
IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

    The End
```

G.2. Apache v2.0

```
                              Apache License
                        Version 2.0, January 2004
                     http://www.apache.org/licenses/

  TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.
   "License" shall mean the terms and conditions for use, reproduction,
   and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by
   the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all
   other entities that control, are controlled by, or are under common
   control with that entity. For the purposes of this definition,
```

```
   "control" means (i) the power, direct or indirect, to cause the
   direction or management of such entity, whether by contract or
   otherwise, or (ii) ownership of fifty percent (50%) or more of the
   outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity
   exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications,
   including but not limited to software source code, documentation
   source, and configuration files.

   "Object" form shall mean any form resulting from mechanical
   transformation or translation of a Source form, including but
   not limited to compiled object code, generated documentation,
   and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or
   Object form, made available under the License, as indicated by a
   copyright notice that is included in or attached to the work
   (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object
   form, that is based on (or derived from) the Work and for which the
   editorial revisions, annotations, elaborations, or other modifications
   represent, as a whole, an original work of authorship. For the purposes
   of this License, Derivative Works shall not include works that remain
   separable from, or merely link (or bind by name) to the interfaces of,
   the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including
   the original version of the Work and any modifications or additions
   to that Work or Derivative Works thereof, that is intentionally
   submitted to Licensor for inclusion in the Work by the copyright owner
   or by an individual or Legal Entity authorized to submit on behalf of
   the copyright owner. For the purposes of this definition, "submitted"
   means any form of electronic, verbal, or written communication sent
   to the Licensor or its representatives, including but not limited to
   communication on electronic mailing lists, source code control systems,
   and issue tracking systems that are managed by, or on behalf of, the
   Licensor for the purpose of discussing and improving the Work, but
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
```

```
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
```

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

G.3. GNU General Public License v1

```
      GNU GENERAL PUBLIC LICENSE
       Version 1, February 1989

 Copyright (C) 1989 Free Software Foundation, Inc.
                59 Temple Place, Suite 330, Boston, MA 02111-1307, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

       Preamble

  The license agreements of most software companies try to keep users
at the mercy of those companies.  By contrast, our General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  The
General Public License applies to the Free Software Foundation's
software and to any other program whose authors commit to using it.
You can use it for your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Specifically, the General Public License is designed to make
sure that you have the freedom to give away or sell copies of free
software, that you receive source code or can get it if you want it,
that you can change the software or use pieces of it in new free
programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of a such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must tell them their rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.
```

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications. Each licensee is addressed as "you".

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this General Public License and to the absence of any warranty; and give any other recipients of the Program a copy of this General Public License along with the Program. You may charge a fee for the physical act of transferring a copy.

2. You may modify your copy or copies of the Program or any portion of it, and copy and distribute such modifications under the terms of Paragraph 1 above, provided that you also do the following:

a) cause the modified files to carry prominent notices stating that you changed the files and the date of any change; and

b) cause the whole of any work that you distribute or publish, that in whole or in part contains the Program or any part thereof, either with or without modifications, to be licensed at no charge to all third parties under the terms of this General Public License (except that you may choose to grant warranty protection to some or all third parties, at your option).

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the simplest and most usual way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this General Public License.

d) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

Mere aggregation of another independent work with the Program (or its derivative) on a volume of a storage or distribution medium does not bring the other work under the scope of these terms.

```
  3. You may copy and distribute the Program (or a portion or derivative of
it, under Paragraph 2) in object code or executable form under the terms of
Paragraphs 1 and 2 above provided that you also do one of the following:

    a) accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of
    Paragraphs 1 and 2 above; or,

    b) accompany it with a written offer, valid for at least three
    years, to give any third party free (except for a nominal charge
    for the cost of distribution) a complete machine-readable copy of the
    corresponding source code, to be distributed under the terms of
    Paragraphs 1 and 2 above; or,

    c) accompany it with the information you received as to where the
    corresponding source code may be obtained.  (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form alone.)

Source code for a work means the preferred form of the work for making
modifications to it.  For an executable file, complete source code means
all the source code for all modules it contains; but, as a special
exception, it need not include source code for modules which are standard
libraries that accompany the operating system on which the executable
file runs, or for standard header files or definitions files that
accompany that operating system.

  4. You may not copy, modify, sublicense, distribute or transfer the
Program except as expressly provided under this General Public License.
Any attempt otherwise to copy, modify, sublicense, distribute or transfer
the Program is void, and will automatically terminate your rights to use
the Program under this License.  However, parties who have received
copies, or rights to use copies, from you under this General Public
License will not have their licenses terminated so long as such parties
remain in full compliance.

  5. By copying, distributing or modifying the Program (or any work based
on the Program) you indicate your acceptance of this license to do so,
and all its terms and conditions.

  6. Each time you redistribute the Program (or any work based on the
Program), the recipient automatically receives a license from the original
licensor to copy, distribute or modify the Program subject to these
terms and conditions.  You may not impose any further restrictions on the
recipients' exercise of the rights granted herein.

  7. The Free Software Foundation may publish revised and/or new versions
of the General Public License from time to time.  Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

Each version is given a distinguishing version number.  If the Program
```

```
specifies a version number of the license which applies to it and "any
later version", you have the option of following the terms and conditions
either of that version or of any later version published by the Free
Software Foundation.  If the Program does not specify a version number of
the license, you may choose any version ever published by the Free Software
Foundation.

  8. If you wish to incorporate parts of the Program into other free
programs whose distribution conditions are different, write to the author
to ask for permission.  For software which is copyrighted by the Free
Software Foundation, write to the Free Software Foundation; we sometimes
make exceptions for this.  Our decision will be guided by the two goals
of preserving the free status of all derivatives of our free software and
of promoting the sharing and reuse of software generally.

       NO WARRANTY

  9. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY
FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.  EXCEPT WHEN
OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES
PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.  THE ENTIRE RISK AS
TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU.  SHOULD THE
PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,
REPAIR OR CORRECTION.

  10. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

       END OF TERMS AND CONDITIONS

 Appendix: How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to humanity, the best way to achieve this is to make it
free software which everyone can redistribute and change under these
terms.

  To do so, attach the following notices to the program.  It is safest to
attach them to the start of each source file to most effectively convey
the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) 19yy  <name of author>
```

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 1, or (at your option)
any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software Foundation,
Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) 19xx name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program `Gnomovision' (a program to direct compilers to make passes
at assemblers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989
Ty Coon, President of Vice
```

That's all there is to it!

G.4. GNU General Public License v2

```
GNU GENERAL PUBLIC LICENSE
   Version 2, June 1991
```

```
 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
     59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

       Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
License is intended to guarantee your freedom to share and change free
software--to make sure the software is free for all its users.  This
General Public License applies to most of the Free Software
Foundation's software and to any other program whose authors commit to
using it.  (Some other Free Software Foundation software is covered by
the GNU Library General Public License instead.)  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if you
distribute copies of the software, or if you modify it.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must give the recipients all the rights that
you have.  You must make sure that they, too, receive or can get the
source code.  And you must show them these terms so they know their
rights.

  We protect your rights with two steps: (1) copyright the software, and
(2) offer you this license which gives you legal permission to copy,
distribute and/or modify the software.

  Also, for each author's protection and ours, we want to make certain
that everyone understands that there is no warranty for this free
software.  If the software is modified by someone else and passed on, we
want its recipients to know that what they have is not the original, so
that any problems introduced by others will not reflect on the original
authors' reputations.

  Finally, any free program is threatened constantly by software
patents.  We wish to avoid the danger that redistributors of a free
program will individually obtain patent licenses, in effect making the
program proprietary.  To prevent this, we have made it clear that any
patent must be licensed for everyone's free use or not licensed at all.
```

```
  The precise terms and conditions for copying, distribution and
modification follow.

      GNU GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License applies to any program or other work which contains
a notice placed by the copyright holder saying it may be distributed
under the terms of this General Public License.  The "Program", below,
refers to any such program or work, and a "work based on the Program"
means either the Program or any derivative work under copyright law:
that is to say, a work containing the Program or a portion of it,
either verbatim or with modifications and/or translated into another
language.  (Hereinafter, translation is included without limitation in
the term "modification".)  Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope.  The act of
running the Program is not restricted, and the output from the Program
is covered only if its contents constitute a work based on the
Program (independent of having been made by running the Program).
Whether that is true depends on what the Program does.

  1. You may copy and distribute verbatim copies of the Program's
source code as you receive it, in any medium, provided that you
conspicuously and appropriately publish on each copy an appropriate
copyright notice and disclaimer of warranty; keep intact all the
notices that refer to this License and to the absence of any warranty;
and give any other recipients of the Program a copy of this License
along with the Program.

You may charge a fee for the physical act of transferring a copy, and
you may at your option offer warranty protection in exchange for a fee.

  2. You may modify your copy or copies of the Program or any portion
of it, thus forming a work based on the Program, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:

    a) You must cause the modified files to carry prominent notices
    stating that you changed the files and the date of any change.

    b) You must cause any work that you distribute or publish, that in
    whole or in part contains or is derived from the Program or any
    part thereof, to be licensed as a whole at no charge to all third
    parties under the terms of this License.

    c) If the modified program normally reads commands interactively
    when run, you must cause it, when started running for such
    interactive use in the most ordinary way, to print or display an
    announcement including an appropriate copyright notice and a
    notice that there is no warranty (or else, saying that you provide
    a warranty) and that users may redistribute the program under
```

```
    these conditions, and telling the user how to view a copy of this
    License.  (Exception: if the Program itself is interactive but
    does not normally print such an announcement, your work based on
    the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Program,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Program, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Program.

In addition, mere aggregation of another work not based on the Program
with the Program (or with a work based on the Program) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may copy and distribute the Program (or a work based on it,
under Section 2) in object code or executable form under the terms of
Sections 1 and 2 above provided that you also do one of the following:

    a) Accompany it with the complete corresponding machine-readable
    source code, which must be distributed under the terms of Sections
    1 and 2 above on a medium customarily used for software interchange; or,

    b) Accompany it with a written offer, valid for at least three
    years, to give any third party, for a charge no more than your
    cost of physically performing source distribution, a complete
    machine-readable copy of the corresponding source code, to be
    distributed under the terms of Sections 1 and 2 above on a medium
    customarily used for software interchange; or,

    c) Accompany it with the information you received as to the offer
    to distribute corresponding source code.  (This alternative is
    allowed only for noncommercial distribution and only if you
    received the program in object code or executable form with such
    an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for
making modifications to it.  For an executable work, complete source
code means all the source code for all modules it contains, plus any
associated interface definition files, plus the scripts used to
control compilation and installation of the executable.  However, as a
special exception, the source code distributed need not include
anything that is normally distributed (in either source or binary
```

form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING,

```
REPAIR OR CORRECTION.

  12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES,
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGES.

       END OF TERMS AND CONDITIONS

     How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software; you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation; either version 2 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program; if not, write to the Free Software
    Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA


Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this
when it starts in an interactive mode:

    Gnomovision version 69, Copyright (C) year  name of author
    Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

  <signature of Ty Coon>, 1 April 1989
  Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

G.5. GNU Lesser General Public License v2.1

GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the

```
    application.  Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole.  If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works.  But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library.  To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License.  (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.)  Do not make any other change in
these notices.

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a
medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
```

compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that

```
    uses the Library", as object code and/or source code, so that the
    user can modify the Library and then relink to produce a modified
    executable containing the modified Library.  (It is understood
    that the user who changes the contents of definitions files in the
    Library will not necessarily be able to recompile the application
    to use the modified definitions.)

    b) Use a suitable shared library mechanism for linking with the
    Library.  A suitable mechanism is one that (1) uses at run time a
    copy of the library already present on the user's computer system,
    rather than copying library functions into the executable, and (2)
    will operate properly with a modified version of the library, if
    the user installs one, as long as the modified version is
    interface-compatible with the version that the work was made with.

    c) Accompany the work with a written offer, valid for at
    least three years, to give the same user the materials
    specified in Subsection 6a, above, for a charge no more
    than the cost of performing this distribution.

    d) If distribution of the work is made by offering access to copy
    from a designated place, offer equivalent access to copy the above
    specified materials from the same place.

    e) Verify that the user has already received a copy of these
    materials or that you have already sent this user a copy.

  For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it.  However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

  It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system.  Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

  7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

    a) Accompany the combined library with a copy of the same work
    based on the Library, uncombined with any other library
    facilities.  This must be distributed under the terms of the
    Sections above.
```

```
    b) Give prominent notice with the combined library of the fact
    that part of it is a work based on the Library, and explaining
    where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute
the Library except as expressly provided under this License.  Any
attempt otherwise to copy, modify, sublicense, link with, or
distribute the Library is void, and will automatically terminate your
rights under this License.  However, parties who have received copies,
or rights, from you under this License will not have their licenses
terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not
signed it.  However, nothing else grants you permission to modify or
distribute the Library or its derivative works.  These actions are
prohibited by law if you do not accept this License.  Therefore, by
modifying or distributing the Library (or any work based on the
Library), you indicate your acceptance of this License to do so, and
all its terms and conditions for copying, distributing or modifying
the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the
Library), the recipient automatically receives a license from the
original licensor to copy, distribute, link with or modify the Library
subject to these terms and conditions.  You may not impose any further
restrictions on the recipients' exercise of the rights granted herein.
You are not responsible for enforcing compliance by third parties with
this License.

  11. If, as a consequence of a court judgment or allegation of patent
infringement or for any other reason (not limited to patent issues),
conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License.  If you cannot
distribute so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you
may not distribute the Library at all.  For example, if a patent
license would not permit royalty-free redistribution of the Library by
all those who receive copies directly or indirectly through you, then
the only way you could satisfy both it and this License would be to
refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices.  Many people have made
generous contributions to the wide range of software distributed
```

through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU

FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.>
Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990
Ty Coon, President of Vice

That's all there is to it!

G.6. GNU Library General Public License v2

```
    GNU LIBRARY GENERAL PUBLIC LICENSE
         Version 2, June 1991

 Copyright (C) 1991 Free Software Foundation, Inc.
                    675 Mass Ave, Cambridge, MA 02139, USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the library GPL.  It is
 numbered 2 because it goes with version 2 of the ordinary GPL.]

       Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Library General Public License, applies to some
specially designated Free Software Foundation software, and to any
other libraries whose authors decide to use it.  You can use it for
your libraries, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
this service if you wish), that you receive source code or can get it
if you want it, that you can change the software or use pieces of it
in new free programs; and that you know you can do these things.

  To protect your rights, we need to make restrictions that forbid
anyone to deny you these rights or to ask you to surrender the rights.
These restrictions translate to certain responsibilities for you if
you distribute copies of the library, or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link a program with the library, you must provide
complete object files to the recipients so that they can relink them
with the library, after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  Our method of protecting your rights has two steps: (1) copyright
the library, and (2) offer you this license which gives you legal
permission to copy, distribute and/or modify the library.
```

Also, for each distributor's protection, we want to make certain that everyone understands that there is no warranty for this free library. If the library is modified by someone else and passed on, we want its recipients to know that what they have is not the original version, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that companies distributing free software will individually obtain patent licenses, thus in effect transforming the program into proprietary software. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License, which was designed for utility programs. This license, the GNU Library General Public License, applies to certain designated libraries. This license is quite different from the ordinary one; be sure to read it in full, and don't assume that anything in it is the same as in the ordinary license.

The reason we have a separate public license for some libraries is that they blur the distinction we usually make between modifying or adding to a program and simply using it. Linking a program with a library, without changing the library, is in some sense simply using the library, and is analogous to running a utility program or application program. However, in a textual and legal sense, the linked executable is a combined work, a derivative of the original library, and the ordinary General Public License treats it as such.

Because of this blurred distinction, using the ordinary General Public License for libraries did not effectively promote software sharing, because most developers did not use the libraries. We concluded that weaker conditions might promote sharing better.

However, unrestricted linking of non-free programs would deprive the users of those programs of all benefit from the free status of the libraries themselves. This Library General Public License is intended to permit developers of non-free programs to use free libraries, while preserving your freedom as a user of such programs to change the free libraries that are incorporated in them. (We have not seen how to achieve this as regards changes in header files, but we have achieved it as regards changes in the actual functions of the Library.) The hope is that this will lead to faster development of free libraries.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, while the latter only works together with the library.

Note that it is possible for a library to be covered by the ordinary

```
General Public License rather than by this special one.

    GNU LIBRARY GENERAL PUBLIC LICENSE
   TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

  0. This License Agreement applies to any software library which
contains a notice placed by the copyright holder or other authorized
party saying it may be distributed under the terms of this Library
General Public License (also called "this License").  Each licensee is
addressed as "you".

  A "library" means a collection of software functions and/or data
prepared so as to be conveniently linked with application programs
(which use some of those functions and data) to form executables.

  The "Library", below, refers to any such software library or work
which has been distributed under these terms.  A "work based on the
Library" means either the Library or any derivative work under
copyright law: that is to say, a work containing the Library or a
portion of it, either verbatim or with modifications and/or translated
straightforwardly into another language.  (Hereinafter, translation is
included without limitation in the term "modification".)

  "Source code" for a work means the preferred form of the work for
making modifications to it.  For a library, complete source code means
all the source code for all modules it contains, plus any associated
interface definition files, plus the scripts used to control compilation
and installation of the library.

  Activities other than copying, distribution and modification are not
covered by this License; they are outside its scope.  The act of
running a program using the Library is not restricted, and output from
such a program is covered only if its contents constitute a work based
on the Library (independent of the use of the Library in a tool for
writing it).  Whether that is true depends on what the Library does
and what the program that uses the Library does.

  1. You may copy and distribute verbatim copies of the Library's
complete source code as you receive it, in any medium, provided that
you conspicuously and appropriately publish on each copy an
appropriate copyright notice and disclaimer of warranty; keep intact
all the notices that refer to this License and to the absence of any
warranty; and distribute a copy of this License along with the
Library.

  You may charge a fee for the physical act of transferring a copy,
and you may at your option offer warranty protection in exchange for a
fee.

  2. You may modify your copy or copies of the Library or any portion
of it, thus forming a work based on the Library, and copy and
distribute such modifications or work under the terms of Section 1
above, provided that you also meet all of these conditions:
```

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also compile or

link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

c) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

d) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

```
If any portion of this section is held invalid or unenforceable under any
particular circumstance, the balance of the section is intended to apply,
and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any
patents or other property right claims or to contest validity of any
such claims; this section has the sole purpose of protecting the
integrity of the free software distribution system which is
implemented by public license practices.  Many people have made
generous contributions to the wide range of software distributed
through that system in reliance on consistent application of that
system; it is up to the author/donor to decide if he or she is willing
to distribute software through any other system and a licensee cannot
impose that choice.

This section is intended to make thoroughly clear what is believed to
be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in
certain countries either by patents or by copyrighted interfaces, the
original copyright holder who places the Library under this License may add
an explicit geographical distribution limitation excluding those countries,
so that distribution is permitted only in or among countries not thus
excluded.  In such case, this License incorporates the limitation as if
written in the body of this License.

  13. The Free Software Foundation may publish revised and/or new
versions of the Library General Public License from time to time.
Such new versions will be similar in spirit to the present version,
but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number.  If the Library
specifies a version number of this License which applies to it and
"any later version", you have the option of following the terms and
conditions either of that version or of any later version published by
the Free Software Foundation.  If the Library does not specify a
license version number, you may choose any version ever published by
the Free Software Foundation.

  14. If you wish to incorporate parts of the Library into other free
programs whose distribution conditions are incompatible with these,
write to the author to ask for permission.  For software which is
copyrighted by the Free Software Foundation, write to the Free
Software Foundation; we sometimes make exceptions for this.  Our
decision will be guided by the two goals of preserving the free status
of all derivatives of our free software and of promoting the sharing
and reuse of software generally.

       NO WARRANTY

  15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO
WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW.
```

```
EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR
OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY
KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE.  THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE
LIBRARY IS WITH YOU.  SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME
THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN
WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU
FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A
FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF
SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH
DAMAGES.

       END OF TERMS AND CONDITIONS

     Appendix: How to Apply These Terms to Your New Libraries

  If you develop a new library, and you want it to be of the greatest
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Library General Public
    License as published by the Free Software Foundation; either
    version 2 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Library General Public License for more details.

    You should have received a copy of the GNU Library General Public
    License along with this library; if not, write to the Free
    Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.
```

```
You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

G.7. Python Software Foundation License v2

```
PYTHON SOFTWARE FOUNDATION LICENSE VERSION 2
--------------------------------------------

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF
hereby grants Licensee a nonexclusive, royalty-free, world-wide
license to reproduce, analyze, test, perform and/or display publicly,
prepare derivative works, distribute, and otherwise use Python
alone or in any derivative version, provided, however, that PSF's
License Agreement and PSF's notice of copyright, i.e., "Copyright (c)
2001, 2002, 2003, 2004 Python Software Foundation; All Rights Reserved"
are retained in Python alone or in any derivative version prepared
by Licensee.

3. In the event Licensee prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of
the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS"
basis.  PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED.  BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
```

```
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee.  This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.
```

Appendix H. Package Licenses

This section enumerates the licenses from all the third party software components of this Roll. A "best effort" attempt has been made to insure the complete and current licenses are listed. In the case of errors or ommisions please contact the maintainer of this Roll. For more information on the licenses of any components please consult with the original author(s) or see the Rocks® CVS repository[1].

H.1. anaconda

GNU General Public License v2

H.2. ant

Apache v2.0

H.2.1. dom

```
This license came from:
http://www.w3.org/Consortium/Legal/copyright-software-19980720

W3C¨ SOFTWARE NOTICE AND LICENSE
Copyright © 1994-2002 World Wide Web Consortium, (Massachusetts Institute of
Technology, Institut National de Recherche en Informatique et en Automatique,
Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

This W3C work (including software, documents, or other related items) is being
provided by the copyright holders under the following license. By obtaining,
using and/or copying this work, you (the licensee) agree that you have read,
understood, and will comply with the following terms and conditions:

Permission to use, copy, modify, and distribute this software and its
documentation, with or without modification,  for any purpose and without fee
or royalty is hereby granted, provided that you include the following on ALL
copies of the software and documentation or portions thereof, including
modifications, that you make:

   1. The full text of this NOTICE in a location viewable to users of the
   redistributed or derivative work.
   2. Any pre-existing intellectual property disclaimers, notices, or terms and
   conditions. If none exist, a short notice of the following form (hypertext
   is preferred, text is permitted) should be used within the body of any
   redistributed or derivative code: "Copyright © [$date-of-software] World
   Wide Web Consortium, (Massachusetts Institute of Technology, Institut
   National de Recherche en Informatique et en Automatique, Keio University).
   All Rights Reserved. http://www.w3.org/Consortium/Legal/"
   3. Notice of any changes or modifications to the W3C files, including the
```

```
   date changes were made. (We recommend you provide URIs to the location from
   which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE
NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED
TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT
THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY
PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR
CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or
publicity pertaining to the software without specific, written prior permission.
Title to copyright in this software and any associated documentation will at all
times remain with copyright holders.
```

H.2.2. sax

```
This license came from: http://www.megginson.com/SAX/copying.html
  However please note future versions of SAX may be covered
  under http://saxproject.org/?selected=pd


This page is now out of date -- see the new SAX site at
http://www.saxproject.org/ for more up-to-date
releases and other information. Please change your bookmarks.


SAX2 is Free!

I hereby abandon any property rights to SAX 2.0 (the Simple API for
XML), and release all of the SAX 2.0 source code, compiled code, and
documentation contained in this distribution into the Public Domain.
SAX comes with NO WARRANTY or guarantee of fitness for any
purpose.

David Megginson, david@megginson.com
2000-05-05
```

H.2.3. xerces

```
The Apache Software License, Version 1.1
```

H.3. coreutils

GNU General Public License v2

H.4. cvs

GNU General Public License v1

H.5. eGenix mx

```
---------------------------------------------------------------------
EGENIX.COM PUBLIC LICENSE AGREEMENT                      VERSION 1.0.0
---------------------------------------------------------------------

1. Introduction

This "License Agreement" is between eGenix.com Software, Skills and
Services GmbH ("eGenix.com"), having an office at Pastor-Loeh-Str. 48,
D-40764 Langenfeld, Germany, and the Individual or Organization
("Licensee") accessing and otherwise using this software in source or
binary form and its associated documentation ("the Software").

2. License

Subject to the terms and conditions of this eGenix.com Public License
Agreement, eGenix.com hereby grants Licensee a non-exclusive,
royalty-free, world-wide license to reproduce, analyze, test, perform
and/or display publicly, prepare derivative works, distribute, and
otherwise use the Software alone or in any derivative version,
provided, however, that the eGenix.com Public License Agreement is
retained in the Software, or in any derivative version of the Software
prepared by Licensee.

3. NO WARRANTY

eGenix.com is making the Software available to Licensee on an "AS IS"
basis.  SUBJECT TO ANY STATUTORY WARRANTIES WHICH CAN NOT BE EXCLUDED,
EGENIX.COM MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED.
BY WAY OF EXAMPLE, BUT NOT LIMITATION, EGENIX.COM MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.
```

4. LIMITATION OF LIABILITY

EGENIX.COM SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE EXCLUSION OR LIMITATION MAY NOT APPLY TO LICENSEE.

5. Termination

This License Agreement will automatically terminate upon a material breach of its terms and conditions.

6. General

Nothing in this License Agreement affects any statutory rights of consumers that cannot be waived or limited by contract.

Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between eGenix.com and Licensee.

If any provision of this License Agreement shall be unlawful, void, or for any reason unenforceable, such provision shall be modified to the extent necessary to render it enforceable without losing its intent, or, if no such modification is possible, be severed from this License Agreement and shall not affect the validity and enforceability of the remaining provisions of this License Agreement.

This License Agreement shall be governed by and interpreted in all respects by the law of Germany, excluding conflict of law provisions. It shall not be governed by the United Nations Convention on Contracts for International Sale of Goods.

This License Agreement does not grant permission to use eGenix.com trademarks or trade names in a trademark sense to endorse or promote products or services of Licensee, or any third party.

The controlling language of this License Agreement is English. If Licensee has received a translation into another language, it has been provided for Licensee's convenience only.

14. Agreement

By downloading, copying, installing or otherwise using the Software, Licensee agrees to be bound by the terms and conditions of this License Agreement.

H.6. FireFox

```
                          MOZILLA PUBLIC LICENSE
                                Version 1.1

                              ---------------

1. Definitions.

     1.0.1. "Commercial Use" means distribution or otherwise making the
     Covered Code available to a third party.

     1.1. "Contributor" means each entity that creates or contributes to
     the creation of Modifications.

     1.2. "Contributor Version" means the combination of the Original
     Code, prior Modifications used by a Contributor, and the Modifications
     made by that particular Contributor.

     1.3. "Covered Code" means the Original Code or Modifications or the
     combination of the Original Code and Modifications, in each case
     including portions thereof.

     1.4. "Electronic Distribution Mechanism" means a mechanism generally
     accepted in the software development community for the electronic
     transfer of data.

     1.5. "Executable" means Covered Code in any form other than Source
     Code.

     1.6. "Initial Developer" means the individual or entity identified
     as the Initial Developer in the Source Code notice required by Exhibit
     A.

     1.7. "Larger Work" means a work which combines Covered Code or
     portions thereof with code not governed by the terms of this License.

     1.8. "License" means this document.

     1.8.1. "Licensable" means having the right to grant, to the maximum
     extent possible, whether at the time of the initial grant or
     subsequently acquired, any and all of the rights conveyed herein.

     1.9. "Modifications" means any addition to or deletion from the
     substance or structure of either the Original Code or any previous
     Modifications. When Covered Code is released as a series of files, a
     Modification is:
          A. Any addition to or deletion from the contents of a file
```

containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.10.1. "Patent Claims" means any patent claim(s), now owned or hereafter acquired, including without limitation, method, process, and apparatus claims, in any patent Licensable by grantor.

1.11. "Source Code" means the preferred form of the Covered Code for making modifications to it, including all modules it contains, plus any associated interface definition files, scripts used to control compilation and installation of an Executable, or source code differential comparisons against either the Original Code or another well known, available Covered Code of the Contributor's choice. The Source Code can be in a compressed or archival form, provided the appropriate decompression or de-archiving software is widely available for no charge.

1.12. "You" (or "Your") means an individual or a legal entity exercising rights under, and complying with all of the terms of, this License or a future version of this License issued under Section 6.1. For legal entities, "You" includes any entity which controls, is controlled by, or is under common control with You. For purposes of this definition, "control" means (a) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (b) ownership of more than fifty percent (50%) of the outstanding shares or beneficial ownership of such entity.

2. Source Code License.

2.1. The Initial Developer Grant.
The Initial Developer hereby grants You a world-wide, royalty-free, non-exclusive license, subject to third party intellectual property claims:

(a) under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work; and

(b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

(c) the licenses granted in this Section 2.1(a) and (b) are

```
      effective on the date Initial Developer first distributes
      Original Code under the terms of this License.

      (d) Notwithstanding Section 2.1(b) above, no patent license is
      granted: 1) for code that You delete from the Original Code; 2)
      separate from the Original Code;  or 3) for infringements caused
      by: i) the modification of the Original Code or ii) the
      combination of the Original Code with other software or devices.

   2.2. Contributor Grant.
   Subject to third party intellectual property claims, each Contributor
   hereby grants You a world-wide, royalty-free, non-exclusive license

      (a)  under intellectual property rights (other than patent or
      trademark) Licensable by Contributor, to use, reproduce, modify,
      display, perform, sublicense and distribute the Modifications
      created by such Contributor (or portions thereof) either on an
      unmodified basis, with other Modifications, as Covered Code
      and/or as part of a Larger Work; and

      (b) under Patent Claims infringed by the making, using, or
      selling of  Modifications made by that Contributor either alone
      and/or in combination with its Contributor Version (or portions
      of such combination), to make, use, sell, offer for sale, have
      made, and/or otherwise dispose of: 1) Modifications made by that
      Contributor (or portions thereof); and 2) the combination of
      Modifications made by that Contributor with its Contributor
      Version (or portions of such combination).

      (c) the licenses granted in Sections 2.2(a) and 2.2(b) are
      effective on the date Contributor first makes Commercial Use of
      the Covered Code.

      (d)    Notwithstanding Section 2.2(b) above, no patent license is
      granted: 1) for any code that Contributor has deleted from the
      Contributor Version; 2)  separate from the Contributor Version;
      3)  for infringements caused by: i) third party modifications of
      Contributor Version or ii)  the combination of Modifications made
      by that Contributor with other software  (except as part of the
      Contributor Version) or other devices; or 4) under Patent Claims
      infringed by Covered Code in the absence of Modifications made by
      that Contributor.

3. Distribution Obligations.

   3.1. Application of License.
   The Modifications which You create or to which You contribute are
   governed by the terms of this License, including without limitation
   Section 2.2. The Source Code version of Covered Code may be
   distributed only under the terms of this License or a future version
   of this License released under Section 6.1, and You must include a
   copy of this License with every copy of the Source Code You
   distribute. You may not offer or impose any terms on any Source Code
```

```
version that alters or restricts the applicable version of this
License or the recipients' rights hereunder. However, You may include
an additional document offering the additional rights described in
Section 3.5.

3.2. Availability of Source Code.
Any Modification which You create or to which You contribute must be
made available in Source Code form under the terms of this License
either on the same media as an Executable version or via an accepted
Electronic Distribution Mechanism to anyone to whom you made an
Executable version available; and if made available via Electronic
Distribution Mechanism, must remain available for at least twelve (12)
months after the date it initially became available, or at least six
(6) months after a subsequent version of that particular Modification
has been made available to such recipients. You are responsible for
ensuring that the Source Code version remains available even if the
Electronic Distribution Mechanism is maintained by a third party.

3.3. Description of Modifications.
You must cause all Covered Code to which You contribute to contain a
file documenting the changes You made to create that Covered Code and
the date of any change. You must include a prominent statement that
the Modification is derived, directly or indirectly, from Original
Code provided by the Initial Developer and including the name of the
Initial Developer in (a) the Source Code, and (b) in any notice in an
Executable version or related documentation in which You describe the
origin or ownership of the Covered Code.

3.4. Intellectual Property Matters
     (a) Third Party Claims.
     If Contributor has knowledge that a license under a third party's
     intellectual property rights is required to exercise the rights
     granted by such Contributor under Sections 2.1 or 2.2,
     Contributor must include a text file with the Source Code
     distribution titled "LEGAL" which describes the claim and the
     party making the claim in sufficient detail that a recipient will
     know whom to contact. If Contributor obtains such knowledge after
     the Modification is made available as described in Section 3.2,
     Contributor shall promptly modify the LEGAL file in all copies
     Contributor makes available thereafter and shall take other steps
     (such as notifying appropriate mailing lists or newsgroups)
     reasonably calculated to inform those who received the Covered
     Code that new knowledge has been obtained.

     (b) Contributor APIs.
     If Contributor's Modifications include an application programming
     interface and Contributor has knowledge of patent licenses which
     are reasonably necessary to implement that API, Contributor must
     also include this information in the LEGAL file.

          (c)    Representations.
     Contributor represents that, except as disclosed pursuant to
     Section 3.4(a) above, Contributor believes that Contributor's
```

Modifications are Contributor’s original creation(s) and/or Contributor has sufficient rights to grant the rights conveyed by this License.

3.5. Required Notices.
You must duplicate the notice in Exhibit A in each file of the Source Code. If it is not possible to put such notice in a particular Source Code file due to its structure, then You must include such notice in a location (such as a relevant directory) where a user would be likely to look for such a notice. If You created one or more Modification(s) You may add your name as a Contributor to the notice described in Exhibit A. You must also duplicate this License in any documentation for the Source Code where You describe recipients’ rights or ownership rights relating to Covered Code. You may choose to offer, and to charge a fee for, warranty, support, indemnity or liability obligations to one or more recipients of Covered Code. However, You may do so only on Your own behalf, and not on behalf of the Initial Developer or any Contributor. You must make it absolutely clear than any such warranty, support, indemnity or liability obligation is offered by You alone, and You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of warranty, support, indemnity or liability terms You offer.

3.6. Distribution of Executable Versions.
You may distribute Covered Code in Executable form only if the requirements of Section 3.1-3.5 have been met for that Covered Code, and if You include a notice stating that the Source Code version of the Covered Code is available under the terms of this License, including a description of how and where You have fulfilled the obligations of Section 3.2. The notice must be conspicuously included in any notice in an Executable version, related documentation or collateral in which You describe recipients’ rights relating to the Covered Code. You may distribute the Executable version of Covered Code or ownership rights under a license of Your choice, which may contain terms different from this License, provided that You are in compliance with the terms of this License and that the license for the Executable version does not attempt to limit or alter the recipient’s rights in the Source Code version from the rights set forth in this License. If You distribute the Executable version under a different license You must make it absolutely clear that any terms which differ from this License are offered by You alone, not by the Initial Developer or any Contributor. You hereby agree to indemnify the Initial Developer and every Contributor for any liability incurred by the Initial Developer or such Contributor as a result of any such terms You offer.

3.7. Larger Works.
You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

```
4. Inability to Comply Due to Statute or Regulation.

     If it is impossible for You to comply with any of the terms of this
     License with respect to some or all of the Covered Code due to
     statute, judicial order, or regulation then You must: (a) comply with
     the terms of this License to the maximum extent possible; and (b)
     describe the limitations and the code they affect. Such description
     must be included in the LEGAL file described in Section 3.4 and must
     be included with all distributions of the Source Code. Except to the
     extent prohibited by statute or regulation, such description must be
     sufficiently detailed for a recipient of ordinary skill to be able to
     understand it.

5. Application of this License.

     This License applies to code to which the Initial Developer has
     attached the notice in Exhibit A and to related Covered Code.

6. Versions of the License.

     6.1. New Versions.
     Netscape Communications Corporation ("Netscape") may publish revised
     and/or new versions of the License from time to time. Each version
     will be given a distinguishing version number.

     6.2. Effect of New Versions.
     Once Covered Code has been published under a particular version of the
     License, You may always continue to use it under the terms of that
     version. You may also choose to use such Covered Code under the terms
     of any subsequent version of the License published by Netscape. No one
     other than Netscape has the right to modify the terms applicable to
     Covered Code created under this License.

     6.3. Derivative Works.
     If You create or use a modified version of this License (which you may
     only do in order to apply it to code which is not already Covered Code
     governed by this License), You must (a) rename Your license so that
     the phrases "Mozilla", "MOZILLAPL", "MOZPL", "Netscape",
     "MPL", "NPL" or any confusingly similar phrase do not appear in your
     license (except to note that your license differs from this License)
     and (b) otherwise make it clear that Your version of the license
     contains terms which differ from the Mozilla Public License and
     Netscape Public License. (Filling in the name of the Initial
     Developer, Original Code or Contributor in the notice described in
     Exhibit A shall not of themselves be deemed to be modifications of
     this License.)

7. DISCLAIMER OF WARRANTY.

     COVERED CODE IS PROVIDED UNDER THIS LICENSE ON AN "AS IS" BASIS,
     WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING,
     WITHOUT LIMITATION, WARRANTIES THAT THE COVERED CODE IS FREE OF
     DEFECTS, MERCHANTABLE, FIT FOR A PARTICULAR PURPOSE OR NON-INFRINGING.
```

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE COVERED CODE IS WITH YOU. SHOULD ANY COVERED CODE PROVE DEFECTIVE IN ANY RESPECT, YOU (NOT THE INITIAL DEVELOPER OR ANY OTHER CONTRIBUTOR) ASSUME THE COST OF ANY NECESSARY SERVICING, REPAIR OR CORRECTION. THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF ANY COVERED CODE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER.

8. TERMINATION.

8.1. This License and the rights granted hereunder will terminate automatically if You fail to comply with terms herein and fail to cure such breach within 30 days of becoming aware of the breach. All sublicenses to the Covered Code which are properly granted shall survive any termination of this License. Provisions which, by their nature, must remain in effect beyond the termination of this License shall survive.

8.2. If You initiate litigation by asserting a patent infringement claim (excluding declatory judgment actions) against Initial Developer or a Contributor (the Initial Developer or Contributor against whom You file such action is referred to as "Participant") alleging that:

(a) such Participant's Contributor Version directly or indirectly infringes any patent, then any and all rights granted by such Participant to You under Sections 2.1 and/or 2.2 of this License shall, upon 60 days notice from Participant terminate prospectively, unless if within 60 days after receipt of notice You either: (i) agree in writing to pay Participant a mutually agreeable reasonable royalty for Your past and future use of Modifications made by such Participant, or (ii) withdraw Your litigation claim with respect to the Contributor Version against such Participant. If within 60 days of notice, a reasonable royalty and payment arrangement are not mutually agreed upon in writing by the parties or the litigation claim is not withdrawn, the rights granted by Participant to You under Sections 2.1 and/or 2.2 automatically terminate at the expiration of the 60 day notice period specified above.

(b) any software, hardware, or device, other than such Participant's Contributor Version, directly or indirectly infringes any patent, then any rights granted to You by such Participant under Sections 2.1(b) and 2.2(b) are revoked effective as of the date You first made, used, sold, distributed, or had made, Modifications made by that Participant.

8.3. If You assert a patent infringement claim against Participant alleging that such Participant's Contributor Version directly or indirectly infringes any patent where such claim is resolved (such as by license or settlement) prior to the initiation of patent infringement litigation, then the reasonable value of the licenses granted by such Participant under Sections 2.1 or 2.2 shall be taken into account in determining the amount or value of any payment or license.

8.4. In the event of termination under Sections 8.1 or 8.2 above, all end user license agreements (excluding distributors and resellers) which have been validly granted by You or any distributor hereunder prior to termination shall survive termination.

9. LIMITATION OF LIABILITY.

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, WHETHER TORT (INCLUDING NEGLIGENCE), CONTRACT, OR OTHERWISE, SHALL YOU, THE INITIAL DEVELOPER, ANY OTHER CONTRIBUTOR, OR ANY DISTRIBUTOR OF COVERED CODE, OR ANY SUPPLIER OF ANY OF SUCH PARTIES, BE LIABLE TO ANY PERSON FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES, EVEN IF SUCH PARTY SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO LIABILITY FOR DEATH OR PERSONAL INJURY RESULTING FROM SUCH PARTY’S NEGLIGENCE TO THE EXTENT APPLICABLE LAW PROHIBITS SUCH LIMITATION. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THIS EXCLUSION AND LIMITATION MAY NOT APPLY TO YOU.

10. U.S. GOVERNMENT END USERS.

The Covered Code is a "commercial item," as that term is defined in 48 C.F.R. 2.101 (Oct. 1995), consisting of "commercial computer software" and "commercial computer software documentation," as such terms are used in 48 C.F.R. 12.212 (Sept. 1995). Consistent with 48 C.F.R. 12.212 and 48 C.F.R. 227.7202-1 through 227.7202-4 (June 1995), all U.S. Government End Users acquire Covered Code with only those rights set forth herein.

11. MISCELLANEOUS.

This License represents the complete agreement concerning subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This License shall be governed by California law provisions (except to the extent applicable law, if any, provides otherwise), excluding its conflict-of-law provisions. With respect to disputes in which at least one party is a citizen of, or an entity chartered or registered to do business in the United States of America, any litigation relating to this License shall be subject to the jurisdiction of the Federal Courts of the Northern District of California, with venue lying in Santa Clara County, California, with the losing party responsible for costs, including without limitation, court costs and reasonable attorneys’ fees and expenses. The application of the United Nations Convention on Contracts for the International Sale of Goods is expressly excluded. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not apply to this License.

12. RESPONSIBILITY FOR CLAIMS.

As between Initial Developer and the Contributors, each party is responsible for claims and damages arising, directly or indirectly, out of its utilization of rights under this License and You agree to work with Initial Developer and Contributors to distribute such responsibility on an equitable basis. Nothing herein is intended or shall be deemed to constitute any admission of liability.

13. MULTIPLE-LICENSED CODE.

Initial Developer may designate portions of the Covered Code as "Multiple-Licensed". "Multiple-Licensed" means that the Initial Developer permits you to utilize portions of the Covered Code under Your choice of the NPL or the alternative licenses, if any, specified by the Initial Developer in the file described in Exhibit A.

EXHIBIT A -Mozilla Public License.

“The contents of this file are subject to the Mozilla Public License Version 1.1 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.mozilla.org/MPL/

Software distributed under the License is distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the License for the specific language governing rights and limitations under the License.

The Original Code is ______________________________________.

The Initial Developer of the Original Code is ________________________.
Portions created by ______________________ are Copyright (C) ______
_______________________. All Rights Reserved.

Contributor(s): ______________________________________.

Alternatively, the contents of this file may be used under the terms of the _____ license (the "[___] License"), in which case the provisions of [______] License are applicable instead of those above. If you wish to allow use of your version of this file only under the terms of the [____] License and not to allow others to use your version of this file under the MPL, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the [___] License. If you do not delete the provisions above, a recipient may use your version of this file under either the MPL or the [___] License."

[NOTE: The text of this Exhibit A may differ slightly from the text of the notices in the Source Code files of the Original Code. You should use the text of this Exhibit A rather than the text found in the Original Code Source Code for Your Modifications.]

```
----------------------------------------------------------------------

AMENDMENTS

The Netscape Public License Version 1.1 ("NPL") consists of the
Mozilla Public License Version 1.1 with the following Amendments,
including Exhibit A-Netscape Public License.  Files identified with
"Exhibit A-Netscape Public License" are governed by the Netscape
Public License Version 1.1.

Additional Terms applicable to the Netscape Public License.
     I. Effect.
     These additional terms described in this Netscape Public
     License -- Amendments shall apply to the Mozilla Communicator
     client code and to all Covered Code under this License.

     II. "Netscape's Branded Code" means Covered Code that Netscape
     distributes and/or permits others to distribute under one or more
     trademark(s) which are controlled by Netscape but which are not
     licensed for use under this License.

     III. Netscape and logo.
     This License does not grant any rights to use the trademarks
     "Netscape", the "Netscape N and horizon" logo or the "Netscape
     lighthouse" logo, "Netcenter", "Gecko", "Java" or "JavaScript",
     "Smart Browsing" even if such marks are included in the Original
     Code or Modifications.

     IV. Inability to Comply Due to Contractual Obligation.
     Prior to licensing the Original Code under this License, Netscape
     has licensed third party code for use in Netscape's Branded Code.
     To the extent that Netscape is limited contractually from making
     such third party code available under this License, Netscape may
     choose to reintegrate such code into Covered Code without being
     required to distribute such code in Source Code form, even if
     such code would otherwise be considered "Modifications" under
     this License.

     V. Use of Modifications and Covered Code by Initial Developer.
          V.1. In General.
          The obligations of Section 3 apply to Netscape, except to
          the extent specified in this Amendment, Section V.2 and V.3.

          V.2. Other Products.
          Netscape may include Covered Code in products other than the
          Netscape's Branded Code which are released by Netscape
          during the two (2) years following the release date of the
          Original Code, without such additional products becoming
          subject to the terms of this License, and may license such
          additional products on different terms from those contained
          in this License.

          V.3. Alternative Licensing.
```

```
        Netscape may license the Source Code of Netscape's Branded
        Code, including Modifications incorporated therein, without
        such Netscape Branded Code becoming subject to the terms of
        this License, and may license such Netscape Branded Code on
        different terms from those contained in this License.

     VI. Litigation.
     Notwithstanding the limitations of Section 11 above, the
     provisions regarding litigation in Section 11(a), (b) and (c) of
     the License shall apply to all disputes relating to this License.

  EXHIBIT A-Netscape Public License.

     "The contents of this file are subject to the Netscape Public
     License Version 1.1 (the "License"); you may not use this file
     except in compliance with the License. You may obtain a copy of
     the License at http://www.mozilla.org/NPL/

     Software distributed under the License is distributed on an "AS
     IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or
     implied. See the License for the specific language governing
     rights and limitations under the License.

     The Original Code is Mozilla Communicator client code, released
     March 31, 1998.

     The Initial Developer of the Original Code is Netscape
     Communications Corporation. Portions created by Netscape are
     Copyright (C) 1998-1999 Netscape Communications Corporation. All
     Rights Reserved.

     Contributor(s): ______________________________________.

     Alternatively, the contents of this file may be used under the
     terms of the _____ license (the "[___] License"), in which case
     the provisions of [______] License are applicable  instead of
     those above.  If you wish to allow use of your version of this
     file only under the terms of the [____] License and not to allow
     others to use your version of this file under the NPL, indicate
     your decision by deleting  the provisions above and replace  them
     with the notice and other provisions required by the [___]
     License.  If you do not delete the provisions above, a recipient
     may use your version of this file under either the NPL or the
     [___] License."
```

H.7. gawk

GNU General Public License v2

H.8. gd

```
Portions copyright 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001,
2002 by Cold Spring Harbor Laboratory. Funded under Grant
P41-RR02188 by the National Institutes of Health.

Portions copyright 1996, 1997, 1998, 1999, 2000, 2001, 2002 by
Boutell.Com, Inc.

Portions relating to GD2 format copyright 1999, 2000, 2001, 2002
Philip Warner.

Portions relating to PNG copyright 1999, 2000, 2001, 2002 Greg
 Roelofs.

Portions relating to gdttf.c copyright 1999, 2000, 2001, 2002 John
Ellson (ellson@lucent.com).

Portions relating to gdft.c copyright 2001, 2002 John Ellson
(ellson@lucent.com).

Portions relating to JPEG and to color quantization copyright 2000,
2001, 2002, Doug Becker and copyright (C) 1994, 1995, 1996, 1997,
1998, 1999, 2000, 2001, 2002, Thomas G. Lane. This software is
based in part on the work of the Independent JPEG Group. See the
file README-JPEG.TXT for more information.

Portions relating to WBMP copyright 2000, 2001, 2002 Maurice
Szmurlo and Johan Van den Brande.

Permission has been granted to copy, distribute and modify gd in
any context without fee, including a commercial application,
provided that this notice is present in user-accessible supporting
documentation.

This does not affect your ownership of the derived work itself, and
the intent is to assure proper credit for the authors of gd, not to
interfere with your productive use of gd. If you have questions,
ask. "Derived works" includes all programs that utilize the
library. Credit must be given in user-accessible documentation.

This software is provided "AS IS." The copyright holders disclaim
all warranties, either express or implied, including but not
limited to implied warranties of merchantability and fitness for a
particular purpose, with respect to this code and accompanying
documentation.

Although their code does not appear in gd, the authors wish to thank
David Koblas, David Rowley, and Hutchison Avenue Software Corporation
for their prior contributions.
```

H.9. graphviz

```
Common Public License Version 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS COMMON PUBLIC
LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM
CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

    a) in the case of the initial Contributor, the initial code and
documentation distributed under this Agreement, and

    b) in the case of each subsequent Contributor:

    i) changes to the Program, and

    ii) additions to the Program;

    where such changes and/or additions to the Program originate from and are
distributed by that particular Contributor. A Contribution 'originates' from a
Contributor if it was added to the Program by such Contributor itself or anyone
acting on such Contributor's behalf. Contributions do not include additions to
the Program which: (i) are separate modules of software distributed in
conjunction with the Program under their own license agreement, and (ii) are not
derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents " mean patent claims licensable by a Contributor which are
necessarily infringed by the use or sale of its Contribution alone or when
combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement,
including all Contributors.

2. GRANT OF RIGHTS

    a) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free copyright license to
reproduce, prepare derivative works of, publicly display, publicly perform,
distribute and sublicense the Contribution of such Contributor, if any, and such
derivative works, in source code and object code form.

    b) Subject to the terms of this Agreement, each Contributor hereby grants
Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed
Patents to make, use, sell, offer to sell, import and otherwise transfer the
Contribution of such Contributor, if any, in source code and object code form.
```

This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

a) it complies with the terms and conditions of this Agreement; and

b) its license agreement:

i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;

ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;

iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and

iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

a) it must be made available under this Agreement; and

b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor’s responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement, including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a cross-claim or counterclaim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed. In addition, if Recipient institutes patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. IBM is the initial Agreement Steward. IBM may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the

```
intellectual property laws of the United States of America. No party to this
Agreement will bring a legal action under this Agreement more than one year
after the cause of action arose. Each party waives its rights to a jury trial in
any resulting litigation.
```

H.10. kudzu

GNU General Public License v2

H.11. libxml2

```
Except where otherwise noted in the source code (e.g. the files hash.c,
list.c and the trio files, which are covered by a similar licence but
with different Copyright notices) all the files are:

 Copyright (C) 1998-2003 Daniel Veillard.  All Rights Reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is fur-
nished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FIT-
NESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  IN NO EVENT SHALL THE
DANIEL VEILLARD BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CON-
NECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Daniel Veillard shall not
be used in advertising or otherwise to promote the sale, use or other deal-
ings in this Software without prior written authorization from him.
```

H.12. libxml2doc

GNU Lesser General Public License v2.1

```
Copyright (C) 2003, 2004, 2005 Paul Boddie <paul@boddie.org.uk>

This library is free software; you can redistribute it and/or
modify it under the terms of the GNU Lesser General Public
License as published by the Free Software Foundation; either
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public
License along with this library; if not, write to the Free Software
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA
```

H.13. mysql

GNU General Public License v2

H.13.1. FLOSS License Exception

```
MySQL FLOSS License Exception

The MySQL AB Exception for Free/Libre and Open Source
Software-only Applications Using MySQL Client Libraries (the
"FLOSS Exception").

Version 0.6, 7 March 2007

Exception Intent

We want specified Free/Libre and Open Source Software (“FLOSS”)
applications to be able to use specified GPL-licensed MySQL client
libraries (the “Program”) despite the fact that not all FLOSS
licenses are compatible with version 2 of the GNU General Public
License (the “GPL”).

Legal Terms and Conditions

As a special exception to the terms and conditions of version 2.0
of the GPL:
```

1. You are free to distribute a Derivative Work that is formed entirely from the Program and one or more works (each, a "FLOSS Work") licensed under one or more of the licenses listed below in section 1, as long as:
   a. You obey the GPL in all respects for the Program and the Derivative Work, except for identifiable sections of the Derivative Work which are not derived from the Program, and which can reasonably be considered independent and separate works in themselves,
   b. all identifiable sections of the Derivative Work which are not derived from the Program, and which can reasonably be considered independent and separate works in themselves,
      i. are distributed subject to one of the FLOSS licenses listed below, and
      ii. the object code or executable form of those sections are accompanied by the complete corresponding machine-readable source code for those sections on the same medium and under the same FLOSS license as the corresponding object code or executable forms of those sections, and
   c. any works which are aggregated with the Program or with a Derivative Work on a volume of a storage or distribution medium in accordance with the GPL, can reasonably be considered independent and separate works in themselves which are not derivatives of either the Program, a Derivative Work or a FLOSS Work.

   If the above conditions are not met, then the Program may only be copied, modified, distributed or used under the terms and conditions of the GPL or another valid licensing option from MySQL AB.

2. FLOSS License List

```
License name Version(s)/Copyright Date
Academic Free License 2.0
Apache Software License 1.0/1.1/2.0
Apple Public Source License 2.0
Artistic license From Perl 5.8.0
BSD license "July 22 1999"
Common Development and Distribution License (CDDL) 1.0
Common Public License 1.0
Eclipse Public License 1.0
GNU Library or "Lesser" General Public License (LGPL) 2.0/2.1
Jabber Open Source License 1.0
MIT license (As listed in file MIT-License.txt) ---
Mozilla Public License (MPL) 1.0/1.1
Open Software License 2.0
OpenSSL license (with original SSLeay license) "2003" ("1998")
PHP License 3.0
Python license (CNRI Python License) ---
Python Software Foundation License 2.1.1
Sleepycat License "1999"
```

University of Illinois/NCSA Open Source License ---
W3C License "2001"
X11 License "2001"
Zlib/libpng License ---
Zope Public License 2.0

Due to the many variants of some of the above licenses, we require that any version follow the 2003 version of the Free Software Foundation's Free Software Definition (http://www.gnu.org/philosophy/free-sw.html) or version 1.9 of the Open Source Definition by the Open Source Initiative (http://www.opensource.org/docs/definition.php).

3. Definitions

    a. Terms used, but not defined, herein shall have the meaning provided in the GPL.
    b. Derivative Work means a derivative work under copyright law.

4. Applicability: This FLOSS Exception applies to all Programs that contain a notice placed by MySQL AB saying that the Program may be distributed under the terms of this FLOSS Exception. If you create or distribute a work which is a Derivative Work of both the Program and any other work licensed under the GPL, then this FLOSS Exception is not available for that work; thus, you must remove the FLOSS Exception notice from that work and comply with the GPL in all respects, including by retaining all GPL notices. You may choose to redistribute a copy of the Program exclusively under the terms of the GPL by removing the FLOSS Exception notice from that copy of the Program, provided that the copy has never been modified by you or any third party.

Appendix A. Qualified Libraries and Packages

The following is a non-exhaustive list of libraries and packages which are covered by the FLOSS License Exception. Please note that this appendix is provided merely as an additional service to specific FLOSS projects wishing to simplify licensing information for their users. Compliance with one of the licenses noted under the "FLOSS license list" section remains a prerequisite.

Package Name	Qualifying License and Version
Apache Portable Runtime (APR)	Apache Software License 2.0

H.14. ncurses

Copyright (c) 1998-2003,2004 Free Software Foundation, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, distribute with modifications, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE ABOVE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name(s) of the above copyright holders shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization.

H.15. numarray

Copyright (C) 2001,2002,2003 Association of Universities for Research in Astronomy (AURA)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of AURA and its representatives may not be used to endorse or promote products derived from this software without specific prior written permission.

H.16. Numeric

Python Software Foundation License v2

This software was originally written by Jim Hugunin when he was at MIT, later at CNRI.

Then Paul Dubois of LLNL became the "keeper". LLNL has released any contributions to it by LLNL personnel for free redistribution. However, many people have contributed to Numpy. It is a classic open source effort that nobody "owns". So, Numerical Python is placed in the SourceForge under the Python license. The LLNL license is in Legal.htm.

-- Paul F. Dubois, Jan. 13, 2000.

H.16.1. LLNL License

```
DISCLAIMER

This software was prepared as an account of work sponsored by an agency of the
United States Government. Neither the United States Government nor the
University of California nor any of their employees, makes any warranty,
express or implied, or assumes any liability or responsibility for the
accuracy, completeness, or usefulness of any information, apparatus, product,
or process disclosed, or represents that its use would not infringe
privately-owned rights. Reference herein to any specific commercial products,
process, or service by trade name, trademark, manufacturer, or otherwise, does
not necessarily constitute or imply its endorsement, recommendation, or favoring
by the United States Government or the University of California. The views and
opinions of authors expressed herein do not necessarily state or reflect those
of the United States Government or the University of California, and shall not
be used for advertising or product endorsement purposes.
```

H.17. perl

GNU General Public License v1, GNU General Public License v2, Artistic License

```
                        Perl Kit, Version 5

         Copyright (C) 1993, 1994, 1995, 1996, 1997, 1998, 1999,
    2000, 2001, 2002, 2003, 2004, 2005, 2006 by Larry Wall and others

                        All rights reserved.

This program is free software; you can redistribute it and/or modify
it under the terms of either:

    a) the GNU General Public License as published by the Free
    Software Foundation; either version 1, or (at your option) any
    later version, or

    b) the "Artistic License" which comes with this Kit.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See either
the GNU General Public License or the Artistic License for more details.

You should have received a copy of the Artistic License with this
Kit, in the file named "Artistic".  If not, I'll be glad to provide one.

You should also have received a copy of the GNU General Public License
along with this program in the file named "Copying". If not, write to the
Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
02111-1307, USA or visit their web page on the internet at
```

```
http://www.gnu.org/copyleft/gpl.html.

For those of you that choose to use the GNU General Public License,
my interpretation of the GNU General Public License is that no Perl
script falls under the terms of the GPL unless you explicitly put
said script under the terms of the GPL yourself.  Furthermore, any
object code linked with perl does not automatically fall under the
terms of the GPL, provided such object code only adds definitions
of subroutines and variables, and does not otherwise impair the
resulting interpreter from executing any standard Perl script.  I
consider linking in C subroutines in this manner to be the moral
equivalent of defining subroutines in the Perl language itself.  You
may sell such an object file as proprietary provided that you provide
or offer to provide the Perl source, as specified by the GNU General
Public License.  (This is merely an alternate way of specifying input
to the program.)  You may also sell a binary produced by the dumping of
a running Perl script that belongs to you, provided that you provide or
offer to provide the Perl source as specified by the GPL.  (The
fact that a Perl interpreter and your code are in the same binary file
is, in this case, a form of mere aggregation.)  This is my interpretation
of the GPL.  If you still have concerns or difficulties understanding
my intent, feel free to contact me.  Of course, the Artistic License
spells all this out for your protection, so you may prefer to use that.
```

H.18. perl tk

Perl License

```
Copyright (c) 1995-2003 Nick Ing-Simmons. All rights reserved.
This package is free software; you can redistribute it and/or
modify it under the same terms as Perl itself, with the exception
of the files in the pTk sub-directory which have separate terms
derived from those of the orignal Tk4.0 sources and/or Tix.

IN NO EVENT SHALL THE AUTHORS OR DISTRIBUTORS BE LIABLE TO ANY PARTY
FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES
ARISING OUT OF THE USE OF THIS SOFTWARE, ITS DOCUMENTATION, OR ANY
DERIVATIVES THEREOF, EVEN IF THE AUTHORS HAVE BEEN ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

THE AUTHORS AND DISTRIBUTORS SPECIFICALLY DISCLAIM ANY WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT.  THIS SOFTWARE
IS PROVIDED ON AN "AS IS" BASIS, AND THE AUTHORS AND DISTRIBUTORS HAVE
NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR
MODIFICATIONS.

See pTk/license.terms for details of this Tk license,
and pTk/Tix.license for the Tix license.
```

H.18.1. pTk

H.18.2. Tix

H.19. pexpect

Python Software Foundation License v2

H.20. phpMyAdmin

Python Software Foundation License v2

H.21. POW

Copyright (c) 2001, 2002, Peter Shannon
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* The name of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

H.22. pygtk

GNU Lesser General Public License v2.1

H.23. python

Python Software Foundation License v2

H.24. rcs

GNU General Public License v2

H.25. readline

GNU General Public License v2

H.26. tidy

```
Copyright (c) 1998-2007 World Wide Web Consortium
(Massachusetts Institute of Technology, European Research
Consortium for Informatics and Mathematics, Keio University).
All Rights Reserved.

This software and documentation is provided "as is," and
the copyright holders and contributing author(s) make no
representations or warranties, express or implied, including
but not limited to, warranties of merchantability or fitness
for any particular purpose or that the use of the software or
documentation will not infringe any third party patents,
copyrights, trademarks or other rights.

The copyright holders and contributing author(s) will not be held
liable for any direct, indirect, special or consequential damages
arising out of any use of the software or documentation, even if
advised of the possibility of such damage.

Permission is hereby granted to use, copy, modify, and distribute
this source code, or portions hereof, documentation and executables,
for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.
2. Altered versions must be plainly marked as such and must
   not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any
   source or altered source distribution.

The copyright holders and contributing author(s) specifically
permit, without fee, and encourage the use of this source code
as a component for supporting the Hypertext Markup Language in
commercial products. If you use this source code in a product,
acknowledgment is not required but would be appreciated.
```

H.27. wget

GNU General Public License v2

Notes

1. http://cvs.rocksclusters.org