How to Install an OSCAR Cluster
Software Version 4.1
Documentation Version 4.1

http://oscar.sourceforge.net/

oscar-users@lists.sourceforge.net

The Open Cluster Group
http://www.openclustergroup.org/

April 18, 2005

Contents

List of Tables

List of Figures

1 Introduction

OSCAR version 4.1 is a snapshot of the best known methods for building, programming, and using clusters. It consists of a fully integrated and easy to install software bundle designed for high performance cluster computing (HPC). Everything needed to install, build, maintain, and use a modest sized Linux cluster is included in the suite, making it unnecessary to download or even install any individual software packages on your cluster.

OSCAR is the first project by the Open Cluster Group. For more information on the group and its projects, visit its website `http://www.OpenClusterGroup.org/`.

This document provides a step-by-step installation guide for system administrators, as well as a detailed explanation of what is happening as you install. Note that this installation guide is specific to OSCAR version 4.1.

1.1 Latest Documentation

Please be sure that you have the latest version of this document. It is possible (and probable!) that newer versions of this document were released on the main OSCAR web site after the software was released. You are *strongly* encouraged to check `http://oscar.sourceforge.net/` for the latest version of these instructions before proceeding. Document versions can be compared by checking their version number and date on the cover page.

1.2 Terminology

A common term used in this document is *cluster*, which refers to a group of individual computers bundled together using hardware and software in order to make them work as a single machine.

Each individual machine of a cluster is referred to as a *node*. Within the OSCAR cluster to be installed, there are two types of nodes: *server* and *client*. A *server* node is responsible for servicing the requests of *client* nodes. A *client* node is dedicated to computation.

An OSCAR cluster consists of one server node and one or more client nodes, where all the client nodes [currently] must have homogeneous hardware. The software contained within OSCAR does support doing multiple cluster installs from the same server, but that process is outside the scope of this guide.

An *OSCAR package* is a set of files that is used to install a software package in an OSCAR cluster. An OSCAR package can be as simple as a single RPM file, or it can be more complex, perhaps including a mixture of RPM and other auxiliary configuration / installation files. OSCAR packages provide the majority of functionality in OSCAR clusters.

OSCAR packages fall into one of three categories:

- *Core packages* are required for the operation of OSCAR itself (mostly involved with the installer).
- *Included packages* are shipped in the official OSCAR distribution. These are usually authored and/or packaged by OSCAR developers, and have some degree of official testing before release.
- *Third party packages* are not included in the official OSCAR distribution; they are "add-ons" that can be unpacked in the OSCAR tree, and therefore installed using the OSCAR installation framework.

1.3 Supported Distributions

OSCAR has been tested to work with several distributions. Table 1 lists each distribution and version and specifies the level of support for each. In order to ensure a successful installation, most users should stick to a distribution that is listed as *Fully supported*.

Distribution and Release	Architecture	Status
Red Hat Linux 9	x86	Fully supported
Red Hat Enterprise Linux 3	x86	Fully supported
Red Hat Enterprise Linux 3	ia64	Fully supported
Fedora Core 2	x86	Fully supported
Mandrakelinux 10.0	x86	Fully supported

Table 1: OSCAR supported distributions

1.4 Minimum System Requirements

The following is a list of minimum system requirements for the OSCAR server node:

- CPU of i586 or above
- A network interface card that supports a TCP/IP stack
- If your OSCAR server node is going to be the router between a public network and the cluster nodes, you will need a second network interface card that supports a TCP/IP stack
- At least 4GB total free space – 2GB under `/` and 2GB under `/var`
- An installed version of Linux, preferably a *Fully supported* distribution from Table 1

The following is a list of minimum system requirements for the OSCAR client nodes:

- CPU of i586 or above
- A disk on each client node, at least 2GB in size (OSCAR will format the disks during the installation)
- A network interface card that supports a TCP/IP stack[1]
- Same Linux distribution and version as the server node
- All clients must have the same architecture (e.g., ia32 vs. ia64)
- Monitors and keyboards may be helpful, but are not required
- Floppy or PXE enabled BIOS

1.5 Document Organization

Due to the complicated nature of putting together a high-performance cluster, it is strongly suggested that even experienced administrators read this document through, without skipping any sections, and then use the detailed installation procedure to install your OSCAR cluster. Novice users will be comforted to know that anyone who has installed and used Linux can successfully navigate through the OSCAR cluster install.

The rest of this document is organized as follows. First, Section 2 tells how to obtain an OSCAR version 4.1 distribution package. Next, the “Release Notes” section (Section 3) that applies to OSCAR version 4.1

[1]Beware of certain models of 3COM cards – not all models of 3COM cards are supported by the installation Linux kernel that is shipped with OSCAR. See the OSCAR web site for more information.

contains some requirements and update issues that need to be resolved before the install. Section 4 provides an overview for the System Installation Suite software package used in OSCAR to perform the bulk of the cluster installation. Section 5 details the cluster installation procedure (the level of detail lies somewhere between "the install will now update some files" and "the install will now replace the string 'xyz' with 'abc' in file `some_file`.")

Finally, Section 6 contains system administration notes about several of the individual packages that are installed by OSCAR. ***This section is a "must read" for all OSCAR system administrators.***

Appendix A covers the topic of network booting client nodes, which is so important that it deserved its own section. Appendix B provides curious users an overview of what really happens during a client install. Appendix C discusses how to install an OSCAR cluster without a DHCP server. Appendix D covers a primer of some security aspects of a Linux cluster. Although not intended to be a comprehensive description of cluster security, it is a good overview for those who know relatively little about system administration and security. Finally, Appendix E is a screen-by-screen walk through of a typical OSCAR installation.

More information is available on the OSCAR web site and archives of the various OSCAR mailing lists. If you have a question that cannot be answered by this document (including answers to common installation problems), be sure to visit the OSCAR web site:

`http://oscar.sourceforge.net/`

2 Downloading an OSCAR Distribution Package

The OSCAR distribution packages can be downloaded from:

`http://oscar.sourceforge.net/`

Note that there are actually three flavors of distribution packages for OSCAR, depending on your bandwidth and installation/development needs:

1. "Regular": All the OSCAR installation material that most users need to install and operate an OSCAR cluster.
2. "Extra Crispy": Same as Regular, except that the SRPMs for most of the RPMs in OSCAR are also included. SRPMs are *not* required for installation or normal operation of an OSCAR cluster. This distribution package is significantly larger than Regular, and is not necessary for most users – only those who are interested in source RPMs need the Extra Crispy distribution. The SRPMs can be found under `packages/*/SRPMS/` directories.
3. "Secret Sauce": This distribution contains *only* the SRPMs for RPMs in OSCAR in the Regular and Extra Crispy distributions (it's essentially (Extra Crispy - Regular)). It is intended only for those who initially downloaded the Regular distribution and later decided that they wanted the SRPMs as well. The Secret Sauce distribution is intended to be expanded over your Regular installation – it will create `packages/*/SRPMS/` directories and populate them with the relevant `.src.rpm` files.

All three distributions can be downloaded from the main OSCAR web page.

3 Release Notes

The following release notes apply to OSCAR version 4.1.

3.1 Release Features

- Red Hat Linux 9, Fedora Core 2 and Mandrakelinux 10.0 support on x86.
- Red Hat Enterprise Linux (RHEL) 3 support on Itanium and x86.
- New RPM dependency finder helps build the server (DepMan/PackMan).
- Ganglia now included in the default package set.
- Torque now included/OpenPBS is now an optional package.
- Multiple bug fixes and Wizard improvements.
- Updated user interface (updated/improved wizard).

3.2 Notes for All Systems

- There is currently a known issue with the Delete Nodes functionality. You may notice error messages like the following when you delete the last node of a cluster:

  ```
  /opt/oscar/packages/sis/scripts/post_clients: illegal or
  non-understood where string <nics.driver_module = >
  /opt/oscar/packages/sis/scripts/post_clients: INTERNAL
  ODA ERROR - failing to convert where expression
  <nics.driver_module = > in oda::delete_records - at
  least one record not deleted in table <nics> in database <oscar>
  /opt/oscar/packages/sis/scripts/post_clients: illegal or
  non-understood where string <nics.gateway = >
  /opt/oscar/packages/sis/scripts/post_clients: INTERNAL
  ODA ERROR - failing to convert where expression
  <nics.gateway = > in oda::delete_records - at least one
  record not deleted in table <nics> in database <oscar>
  /opt/oscar/packages/sis/scripts/post_clients: illegal or
  non-understood where string <nics.assignment_method = >
  ```

  This is not 'harmful' but will leave the ODA database inconsistent with the SIS database. The information on network interfaces is currently not used therefore it is not really damaging to your installation. This will be fixed in the next release.

- In Step 8 <**`Test Cluster Setup`**>, some tests may fail for the first time but subsequent re-runs of this step will indicate that all the tests succeeded. If APItests are failing, it may be because the OSCAR Wizard did not have the correct environment variables set. The way around this problem is to quit the Wizard, open another terminal, bring up the Wizard and re-run the tests. All tests (including APItests) should pass if the cluster is configured correctly.

- Each package in OSCAR has its own installation and release notes. See Section 6 for additional release notes.

- All nodes must have a hostname other than "`localhost`" that does not contain any underscores ("_") or periods "`.`". Some distributions complicate this by putting a line such as as the following in /etc/hosts:

  ```
    127.0.0.1   localhost.localdomain   localhost
    yourhostname.yourdomain    yourhostname
  ```

  If this occurs the file should be separated as follows:

  ```
  127.0.0.1 localhost.localdomain localhost
  192.168.0.1 yourhostname.yourdomain yourhostname
  ```

- A domain name must be specified for the client nodes when defining them.
- Due to some distribution portability issues, OSCAR currently installs a "compatibility" (`python2--compat-1.0-1`) RPM to resolve the Python2 prerequisite that is slightly different across different Linux distributions. Also see the file `packages/c3/RPMS/NOTE.python2`.
- In some cases, the test window that is opened from the OSCAR wizard may close suddenly when there is a test failure. If this happens, run the test script, `testing/test_cluster`, manually in a shell window to diagnose the problem.
- Although OSCAR can be installed on pre-existing server nodes, it is typically easiest to use a machine that has a new, fresh install of a distribution listed in Table 1 *with no updates installed*. If the updates are installed, there may be conflicts in RPM requirements. It is recommended to install RedHat updates *after* the initial OSCAR installation has completed.
- The following benign warning messages will appear multiple times during the OSCAR installation process:

  ```
  awk: cmd. line:2: fatal: cannot open file '/etc/fstab'
    for reading (No such file or directory)

  rsync_stub_dir: no such variable at ...

  Use of uninitialized value in pattern match (m//) at
  /usr/lib/perl5/site_perl/oda.pm ...
  ```

  It is safe to ignore these messages.
- The OSCAR installer will install the MySQL package on the server node if it is not already installed. A random password will be automatically generated for the oscar user to access the oscar database. This password will be stored in the file `/etc/odapw`. It should not be needed by other users.
- The OSCAR installer GUI provides little protection for user mistakes. If the user executes steps out of order, or provides erroneous input, Bad Things may happen. Users are strongly encouraged to closely follow the instructions provided in this document.

- The OSCAR installer GUI currently does not support deleting a node and adding the same node back *in the same session*. If you wish to delete a node and then add it back, you must delete the node, close the OSCAR installer GUI, launch the OSCAR installer GUI again, and then add the node.
- If `ssh` produces warnings when logging into the compute nodes from the OSCAR head node, the C3 tools (e.g., `cexec`) may experience difficulties. For example, if you use `ssh` to login in to the OSCAR head node from a terminal that does not support X windows and then try to run `cexec`, you might see a warning message in the `cexec` output:

  ```
  Warning: No xauth data; using fake authentication data for
  X11 forwarding.
  ```

  Although this is only a warning message from `ssh`, `cexec` may interpret it as a fatal error, and not run across all cluster nodes properly (e.g., the <**`Install/Uninstall Packages`**> button will likely not work properly).

  Note that this is actually an `ssh` problem, not a C3 problem. As such, you need to eliminate any warning messages from ssh (more specifically, eliminate any output from `stderr`). In the example above, you can tell the C3 tools to use the "`-x`" switch to `ssh` in order to disable X forwarding:

  ```
  # export C3_RSH='ssh -x'
  # cexec uptime
  ```

  The warnings about `xauth` should no longer appear (and the <**`Install/Uninstall Packages`**> button should work properly).
- The <**`Cancel`**> button in the <**`Install/Uninstall Package`**> step does not work properly; if any packages are selected to be installed or uninstalled, clicking the <**`Cancel`**> button still triggers the execution of the package installer/uninstaller. This will be fixed in a future release. The same behavior occurs if you close the window via the window manager's "close" functionality.

  Note that if you do not select any additional packages to install/uninstall, nothing will run (as expected).
- The SIS multicast facility (Flamethrower) is "experimentally" supported. If you are having problems with multicast and would like to experiment please check the `oscar-users` and/or `sisuite-users` mailing lists for tips.
- In the <**`Setup Networking`**> step, MACs can only be saved in /root which is the default directory.
- The man pages for the Torque package will not be available in a default installation because /opt/pbs/man is missing from MANPATH. They should appear if this is added by hand eg:

  ```
  # export MANPATH=$MANPATH:/opt/pbs/man
  ```
- *FutureWarning* message during APItests on Python2.3 based systems. The following is a warning message about the for the version of TwistedMatrix used by the APItest tool. It is only a warning and can be ignored.

```
Running Installation tests for pvm
/usr/lib/python2.3/site-packages/twisted/internet/defer.py:398:
FutureWarning: hex()/oct() of negative int will return
a signed string in Python 2.4 and up return "<%s at %s>"
% (cname, hex(id(self)))
```

3.3 Red Hat Linux 9 Notes

There are a few issues that may crop up when using OSCAR on Red Hat 9. The following items highlight these issues.

- Deselecting Pfilter causes the image creation to fail. This is due to a dependency with IPtables and when Pfilter is not selected the IPtables RPM is not listed in the node (image) rpmlist. The simple fix is to add “iptables” to the Red Hat 9 rpmlist if you are not installing Pfilter on the compute nodes.
- The RPM system has been updated with this Red Hat release. The OSCAR install process will likely display several warnings due to unsigned RPMS. These warnings can be ignored.
- In some OSCAR pre-release testing, RPM would hang during the building of a client image (Section 5.7). This is a documented bug in the version of RPM that ships with Redhat 9; it is not a problem with OSCAR. The procedure that was used to remedy this situation is outlined below (excerpts taken from `http://www.rpm.org/hintskinks/repairdb-2003-06/`):
  - If RPM hangs at any point (e.g., building the client image) – first ensure that it really has hung and just isn’t taking a long, long time to complete. Typical indications that it has genuinely hung include: the disk is not running and load goes down to 0 (or nearly 0) and stays there.
  - Then do a `ps` and find the PID of the `rpm` process:

    ```
    # ps -eadf | grep rpm | grep -v grep
    ...output...
    # kill <PID_of_RPM>
    ```

  - This will probably not kill the process (it’s likely to be in a state where it is ignoring signals), but it should be tried anyway – this would allow `rpm` to exit cleanly. If `rpm` does exit cleanly, jump down to the last step in this procedure.
  - If `rpm` does not exit within a short period of time, `kill -9 <PID_of_RPM>`. This guarantees that `rpm` will not exit cleanly, but in this case, it’s ok. Now, do the following:
    1. Save a copy of the RPM database (just to be safe):

       ```
       # cd /var/lib
       # tar zcvf /tmp/rpmdb.tar.gz rpm
       ```

    2. Delete any existing RPM database locks:

       ```
       # cd /var/lib/rpm
       # rm -f __db*
       ```

    3. Rebuild the RPM database:

       ```
       # rpm -vv --rebuilddb
       ```

  - Now re-run the OSCAR step that hung. If RPM hangs again, repeat these steps to un-hang it. Testing has shown that it may be necessary to repeat these steps multiple times in order to get a successful RPM run.

3.4 Red Hat Enterprise Linux 3 Notes

Currently we only support Red Hat Enterprise Linux 3 Update 2 and Update 3. Gold and Update 4 are not supported.

If you are installing a version of Red Hat Enterprise Linux 3 that does not provide a MySQL Server RPM (eg. WS), please refer to the first item ’mysql-server’.

If you are installing on Red Hat Enterprise Linux 3 Update 2, please refer to the second item ’Red Hat Enterprise Linux 3 (Update 2) rpmlist’.

The rest of the notes are only relevant if you are installing on ia64 (Itanium) hardware.

- mysql-server

  BEFORE YOU BEGIN: you will need to obtain a mysql-server RPM and put it into

  `/tftpboot/rpm`

  The easiest way is to get the SRPM and rebuild it(i386 users should replace ia64 with i386),

  ```
  rpmbuild --rebuild mysql-3.23.58-1.src.rpm
  cp /usr/src/redhat/RPMS/ia64/mysql-server-3.23.58-1.ia64.rpm /tftpboot/rpm
  ```

  MySQL v3.23.58-1 is the version that came with Red Hat Enterprise Linux 3, Update 3; if you cannot find that particular version, please make sure that you copy all the rebuilt MySQL RPMs (including the server RPM). For example, if the version you found is v3.23.58-2.3:

  ```
  rpmbuild --rebuild mysql-3.23.58-2.3.src.rpm
  cp /usr/src/redhat/RPMS/ia64/mysql*-3.23.58-2.3.ia64.rpm /tftpboot/rpm
  ```

  Note: The key here is to keep the MySQL versions consistent.

- Red Hat Enterprise Linux 3 (Update 2) rpmlist

  IN OSCAR WIZARD STEP 4: if you’re running Update 2, you will need to manually select the correct rpmlist before generating the client image. Select the appropriate file for your architecture

  `/opt/oscar/oscarsample/redhat-3asU2-i386.rpmlist`
  or
  `/opt/oscar/oscarsample/redhat-3asU2-ia64.rpmlist`

- `systemconfig.conf` on ia64

  AFTER OSCAR WIZARD STEP 4: you will need to add an INITRD entry to the image file

  `/var/lib/systemimager/images/oscarimage/etc/systemconfig/systemconfig.conf`

  After the modification, the kernel section should look like this:

  ```
  [KERNEL0]
    PATH = /boot/efi//EFI/redhat/vmlinuz-2.4.21-20.EL
    INITRD = /boot/efi//EFI/redhat/initrd-2.4.21-20.EL.img
    LABEL = 2.4.21-20.EL
  ```

where 2.4.21-20.EL is the kernel for Red Hat Enterprise Linux 3 (Update 3) you should substitute your kernel version if you're not running Update 3.

Note: Note carefully the *DOUBLE SLASH* in the PATH and INITRD lines!

- SCSI and network on ia64

  AFTER OSCAR WIZARD STEP 4: you may also need to add a Hardware section with SCSI and network drivers. In the image file

  `/var/lib/systemimager/images/oscarimage/etc/systemconfig/systemconfig.conf`

  For an Intel SR870BH2, the hardware section of this file would look like:

  ```
  [HARDWARE]
    ORDER = e1000 e1000 mptscsih mptbase scsi_mod
  ```

- USB on ia64

  AFTER OSCAR WIZARD STEP 4: if you need to use the keyboard on a USB-only system, like the Intel SR870BH2, you need to add the USB controller to the image file

  `/var/lib/systemimager/images/oscarimage/etc/modules.conf`

  For example,

  ```
  echo alias usb-controller usb-uhci >> \
    /var/lib/systemimager/images/oscarimage/etc/modules.conf
  ```

3.5 Fedora Core 2 Notes

- SIS currently does not fully support 2.6 kernel, the way we make it work right now is to run a script which generate `/etc/modprobe.conf` from `/etc/modules.conf` - this does not work 100% of the time. If you can deploy an image to your client nodes but are having problems booting, chances are this is the issue. To get around this problem, copy a working `modprobe.conf` for your client nodes to the image directory `/var/lib/systemimager/images/oscarimage/etc/`. Re-image the nodes and it should work. One quick way of getting the correct `modprobe.conf` is to boot a node with the Fedora Core 2 CD 1 and run the Rescue mode. You will then be able to find `modprobe.conf` in /tmp.

3.6 Mandrakelinux 10.0 Notes

- The version of Mandrakelinux supported is 10.0 Official which is a 3CD set. While it may be possible to install OSCAR with 10.0 Community, it is not officially supported.

- During Step 3, <**`Install OSCAR Server Packages`**>, the RPM package `kernel-enterprise` will be selected to be installed on the headnode. As a result, this kernel will be used upon subsequent reboot.

- While Mandrakelinux 10.0 supports 2.4 kernel, OSCAR only supports 2.6 kernel - this is explicitly specified in `oscarsamples/mandrake-10.0-i386.rpmlist`.

- The `tftp-server` RPM which came with Mandrakelinux 10.0 expects the pxelinux boot files to reside in `/var/lib/tftpboot` but OSCAR puts them in `/tftpboot`. To get around this issue a symbolic link is created from `/var/lib/tftpboot` to `/tftpboot`.

4 Overview of System Installation Suite (SIS)

The first question you may have is “what is SIS?” The System Installation Suite (SIS) is a cluster installation tool developed by the collaboration of the IBM Linux Technology Center and the SystemImager team. SIS was chosen to be the installation mechanism for OSCAR for multiple reasons:

- SIS is a high-quality, third party, open source product that works well in production environments
- SIS does not require the client nodes to already have Linux installed
- SIS maintains a database containing installation and configuration information about each node in the cluster
- SIS uses RPM as a standard for software installation
- SIS supports heterogenous hardware and software installation (although this feature is not [yet] used by OSCAR)

In order to understand some of the steps in the upcoming install, you will need knowledge of the main concepts used within SIS. The first concept is that of an *image*. In SIS, an *image* is defined for use by the cluster nodes. This image is a copy of the operating system files stored on the server. The client nodes install by replicating this image to their local disk partitions. Another important concept from SIS is the client definition. A SIS client is defined for each of your cluster nodes. These client definitions keep track of the pertinent information about each client. The server node is responsible for creating the cluster information database and for servicing client installation requests. The information that is stored for each client includes:

- IP information such as hostname, IP address, route.
- Image name.

Each of these pieces of information will be discussed further as part of the detailed install procedure.

For additional information on the concepts in SIS and how to use it, you should refer to the `SIS(1)` man page. In addition, you can visit the SIS web site at `http://sisuite.org/` for recent updates.

5 Detailed Cluster Installation Procedure

All actions specified below should be performed by the `root` user on the server node unless noted otherwise. Note that if you login as a regular user and use the `su` command to change to the `root` user, you *must* use “`su -`” to get the full `root` environment. Using “`su`” (with no arguments) is not sufficient, and will cause obscure errors during an OSCAR installation.

Note that all the steps below are mandatory unless explicitly marked as optional.

5.1 Server Installation and Configuration

During this phase, you will prepare the machine to be used as the server node in the OSCAR cluster.

5.1.1 Install Linux on the server machine

If you have a machine you want to use that already has Linux installed, ensure that it meets the minimum requirements as listed in Section 1.4. If it does, you may skip ahead to Section 5.1.2.

It should be noted that OSCAR is only supported on the distributions listed in Table 1 (page 7). As such, use of distributions other than those listed will likely require some porting of OSCAR, as many of the scripts and software within OSCAR are dependent on those distributions.

When installing Linux, it is not necessary to perform a "custom" install since OSCAR will usually install all the software on which it depends. The main Linux installation requirement is that some X windowing environment such as GNOME or KDE must be installed. Typically, a "Workstation" install yields a sufficient installation for OSCAR to install successfully.

It is best to *not* install distribution updates after you install Linux; doing so may disrupt some of OSCAR's RPM dependencies. Instead, install OSCAR first, and then install the distribution updates.

5.1.2 Disk space and directory considerations

OSCAR has certain requirements for server disk space. Space will be needed to store the Linux RPMs and to store the images. The RPMs will be stored in `/tftpboot/rpm`. 2GB is usually enough to store the RPMs. The images are stored in `/var/lib/systemimager` and will need approximately 2GB per image. Although only one image is required for OSCAR, you may want to create more images in the future.

If you are installing a new server, it is suggested that you allow for 4GB in both the `/` (which contains `/tftpboot`) and `/var` filesystems when partitioning the disk on your server.

If you are using an existing server, you will need to verify that you have enough space on the disk partitions. Again 4GB of free space is recommended under each of `/` and `/var`.

You can check the amount of free space on your drive's partitions by issuing the command `df -h` in a terminal. The result for each file system is located below the `Available` column heading. If your root (`/`) partition has enough free space, enter the following command in a terminal:

```
# mkdir -p /tftpboot/rpm
```

If your root partition does not have enough free space, create the directories on a different partition that does have enough free space and create symbolic links to them from the root (`/`) directory. For example, if the partition containing `/usr` contains enough space, you could do so by using the following commands:

```
# mkdir -p /usr/tftpboot/rpm
# ln -s /usr/tftpboot /tftpboot
```

The same procedure should be repeated for the `/var/lib/systemimager` subdirectory.

5.1.3 Download a copy of OSCAR and unpack on the server

If you are reading this, you probably already have a copy of an OSCAR distribution package. If not, go to `http://oscar.sourceforge.net/` and download the latest OSCAR Regular or Extra Crispy distribution package (see Section 2, page 8). Ensure that you have the latest documentation (later documentation may be available on the OSCAR web site than in the OSCAR distribution package).

Place the OSCAR distribution package in a directory such as `root`'s home directory on the server node. Although there is no required installation directory (note that you may not use the directory `/usr/local/oscar`, `/opt/oscar`, `/var/lib/oscar`, or `/var/cache/oscar` – they are reserved for use by OSCAR), the rest of these instructions will assume that you downloaded the OSCAR distribution package to `root`'s home directory.

Do **not** unpack the tarball on a Windows-based machine and copy the directories over to the server, as this will convert all the scripts to "DOS" format and will render them useless under Linux.

Open a command terminal and issue the following commands to unpack the OSCAR distribution package:

```
# cd
# tar zxf <filename>
```

Where <`filename`> is either `oscar-4.1.tar.gz` (regular distribution) or `oscar-including--srpms-4.1.tar.gz` (extra crispy distribution).

Directory	Contents
`~/oscar-4.1/`	the base OSCAR directory
`~/oscar-4.1/COPYING`	GNU General Public License v2
`~/oscar-4.1/dist`	distribution scripts for configure/install
`~/oscar-4.1/doc`	OSCAR documentation directory
`~/oscar-4.1/images`	auxiliary images used in the GUI
`~/oscar-4.1/install_cluster`	main installation script
`~/oscar-4.1/lib`	auxiliary library routines
`~/oscar-4.1/oscarsamples`	sample configuration files
`~/oscar-4.1/packages`	RPM and installation files for the OSCAR packages
`~/oscar-4.1/README`	text README document
`~/oscar-4.1/scripts`	contains scripts that do most of the work
`~/oscar-4.1/share`	more auxiliary helper files
`~/oscar-4.1/testing`	contains OSCAR cluster test helper scripts
`~/oscar-4.1/VERSION`	file containing the OSCAR version number

Table 2: OSCAR distribution package file and directory layout.

5.1.4 Configure and Install OSCAR

Starting with OSCAR 2.3, after unpacking the tarball you will need to configure and install OSCAR. The configure portion sets up the release to be permanently installed on the system in `/opt/oscar` (default location). The `--prefix=`*`ALT-DIR`* flag can be used to configure and install OSCAR into an alternate directory.

The first step is to run the `configure` script, which is provided in the top-level directory of the OSCAR release.

```
# cd /root/oscar-4.1
# ./configure
```

Once the configure script successfully completes you are ready to actually install OSCAR on the server. At this point no changes have been made to the system beyond the `oscar-4.1` directory itself. Running the following will copy the files to the system. The files copied will include the base OSCAR toolkit as well as startup scripts for the `profile.d/` area. The startup scripts add OSCAR to your path and set environment variables like `OSCAR_HOME`.

```
# make install
```

At this point you will be ready to change to either the default `/opt/oscar` directory or whatever path was use with the `--prefix` flag to perform the installation steps discussed in Section 5.2.

In the remainder of this document, the variable `$OSCAR_HOME` will be used in place of the directory you installed OSCAR to – by default this is `/opt/oscar`.

5.1.5 Configure the ethernet adapter for the cluster

Assuming you want your server to be connected to both a public network and the private cluster subnet, you will need to have two ethernet adapters installed in the server. This is the preferred OSCAR configuration because exposing your cluster may be a security risk and certain software used in OSCAR (such as DHCP) may conflict with your external network.

Once both adapters have been physically installed in the server node, you need to configure them.[2] Any network configurator is sufficient; popular applications include `neat`, `netcfg`, or a text editor.

The following major requirements need to be satisfied:

Hostname. Most Linux distributions default to the hostname "`localhost`" (or "`localhost.localdomain`"). This must be changed in order to successfully install OSCAR – choose another name that does not include any underscores ("_"). This may involve editing /etc/hosts by hand as some distributions hide the lines involving "`localhost`" in their graphical configuration tools. Do not remove all reference to "`localhost`" from /etc/hosts as this will cause no end of problems.

For example if your distribution automatically generates the /etc/hosts file:

```
127.0.0.1    localhost.localdomain    localhost
yourhostname.yourdomain    yourhostname
```

This file should be separated as follows:

```
127.0.0.1 localhost.localdomain localhost
192.168.0.1 yourhostname.yourdomain yourhostname
```

Additional lines may be needed if more than one network adapter is present.

Public adapter. This is the adapter that connects the server node to a public network. Although it is not required to have such an adapter, if you do have one, you must configure it as appropriate for the public network (you may need to consult with your network administrator).

Private adapter. This is the adapter connected to the TCP/IP network with the rest of the cluster nodes. This adapter must be configured as follows:

- Use a private IP address[3]
- Use an appropriate netmask[4]

[2] Beware of certain models of 3COM network cards. See footnote 1 on page 7.

[3] There are three private IP address ranges: 10.0.0.0 to 10.255.255.255; 172.16.0.0 to 172.32.255.255; and 192.168.0.0 to 192.168.255.255. Additional information on private intranets is available in RFC 1918. You should not use the IP addresses 10.0.0.0 or 172.16.0.0 or 192.168.0.0 for the server. If you use one of these addresses, the network installs of the client nodes will fail.

[4] A class C netmask of 255.255.255.0 should be sufficient for most OSCAR clusters.

- Ensure that the interface is activated at boot time
- Set the interface control protocol to "none"

Now reboot the server node to ensure that all the changes are propagated to the appropriate configuration files. To confirm that all ethernet adapters are in the "up" state, once the machine has rebooted, open another terminal window and enter the following command:

```
# ifconfig -a
```

You should see `UP` as the first word on the third line of output for each adapter. If not, there is a problem that you need to resolve before continuing. Typically, the problem is that the wrong module is specified for the given device. Try using the network configuration utility again to resolve the problem.

5.1.6 Copy distribution installation RPMs to `/tftpboot/rpm`

In this step, you need to copy all the RPMs included with your Linux distribution into the `/tftpboot/rpm` directory. Most distributions have more RPMs than are generally necessary for installation, it is important that the RPMs from all the installation discs are included. When each CD is inserted, Linux usually automatically makes the contents of the CD be available in the `/mnt/cdrom` directory (you may need to execute a command such as "`mount /mnt/cdrom`" if the CD does not mount automatically).

For each CD, locate the directory that contains the RPMs. In Fedora Core distributions, the RPMs are located in the `Fedora/RPMS` directory (i.e., `/mnt/cdrom/Fedora/RPMS`). Copy the RPMS into the `/tftpboot/rpm` directory with a command such as [5]:

```
# cp /mnt/cdrom/RedHat/RPMS/*.rpm /tftpboot/rpm
```

Be sure to repeat the above process for *all* CDs. After using each CD you will have to unmount it from the local file system and eject it by issuing these commands:

```
# cd
# eject cdrom
```

5.2 Launching the OSCAR Installer

Change directory to the top-level OSCAR directory and start the OSCAR install wizard:

```
# cd $OSCAR_HOME
# ./install_cluster <device>
```

In the above command, substitute the device name (e.g., *eth1*) in place of `<device>` for your server's private network ethernet adapter. The script will execute some setup / configuration steps, including (but not limited to):

1. installs prerequisite packages on the server
2. copies OSCAR RPMs to `/tftpboot/rpm`
3. installs all OSCAR server RPMs
4. updates `/etc/hosts` with OSCAR aliases
5. updates `/etc/exports`

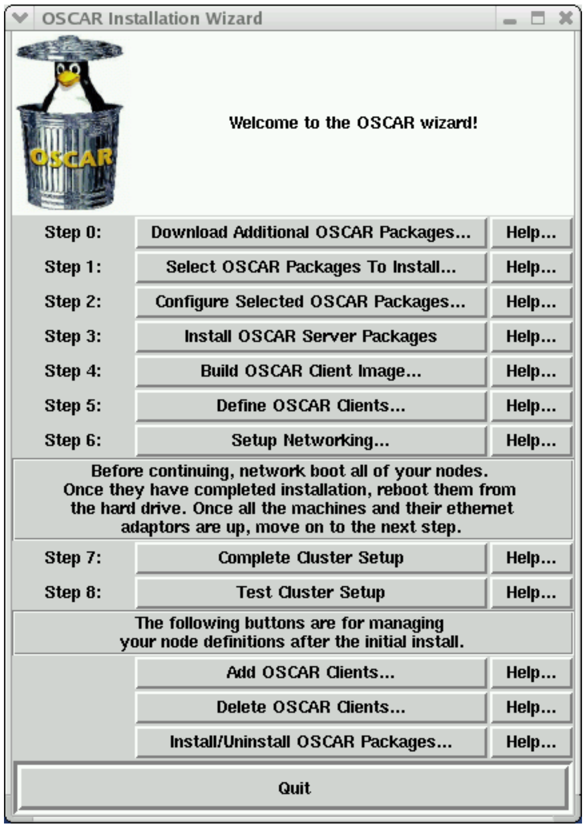
[5]It may be helpful to use `rsync` for these mass copies to ensure the integrity of the data (at a cost of longer transfer times). For example replace the `cp` command in the example with `rsync -cav --stats`.

6. adds OSCAR paths to `/etc/profile`
7. updates system startup (`/etc/rc.d/init.d`) scripts
8. restarts affected services

A lot of output will be displayed in the console window where you invoked `install_cluster`. This reflects normal operational output from the various installation commands that OSCAR executes. The output is also saved in the file `oscarinstall.log` for later reference (particularly if something goes wrong during during the installation).

After the steps listed above have successfully finished, the OSCAR installation wizard GUI will automatically be launched.

The wizard, as shown in Figure 1, is provided to guide you through the rest of the cluster installation. To use the wizard, you will complete a series of steps, with each step being initiated by the pressing of a button on the wizard. Do not go on to the next step until the instructions say to do so, as there are times when you may need to complete an action outside of the wizard before continuing on with the next step. For each step, there is also a <**`Help`**> button located directly to the right of the step button. When pressed, the <**`Help`**> button displays a message box describing the purpose of the step.


OSCAR Installation Wizard

Welcome to the OSCAR wizard!

Before continuing, network boot all of your nodes. Once they have completed installation, reboot them from the hard drive. Once all the machines and their ethernet adaptors are up, move on to the next step.

The following buttons are for managing your node definitions after the initial install.

OSCAR
Step 0: Download Additional OSCAR Packages... Help...
Step 1: Select OSCAR Packages To Install... Help...
Step 2: Configure Selected OSCAR Packages... Help...
Step 3: Install OSCAR Server Packages Help...
Step 4: Build OSCAR Client Image... Help...
Step 5: Define OSCAR Clients... Help...
Step 6: Setup Networking... Help...
Step 7: Complete Cluster Setup Help...
Step 8: Test Cluster Setup Help...
Add OSCAR Clients... Help...
Delete OSCAR Clients... Help...
Install/Uninstall OSCAR Packages... Help...
Quit

Figure 1: OSCAR Wizard.

5.3 Downloading Additional OSCAR Packages

Note: This step is optional.

The first step of the Wizard, "Step 0", enables you to download additional packages. The OSCAR Package Downloader (OPD) is a unified method of downloading OSCAR packages and inserting them in the OSCAR installation hierarchy so that they will be installed during the main OSCAR installation process. The Wizard uses a GUI frontend to OPD affectionately known as *OPDer*. The addition of this frontend limits the need for direct access to the command-line OPD tool directly.[6] If you would like to add additional repository URLs for testing purposes, you can do this by accessing the <**`File`**> menu and then choosing <**`Additional Repositories...`**>. The remainder of this sub-section describes the underlying OPD tool.

The command-line OPD can be executed outside of the GUI Wizard from the top-level OSCAR directory with the following command:

```
# cd $OSCAR_HOME
# ./scripts/opd
```

OPD can use either the `wget` command or the built-in Perl LWP for downloading files. By default, `wget` will be used if it can be found. However, the use of LWP may be forced if the argument "`--lwp`" is given on the command line. LWP may be used in order to utilize a proxy, for example. See the `LWP::UserAgent` documentation for details on how to use proxies through LWP.

OPD also will make use of the "`http_proxy`" environment variable if it is set, as will many other shell applications. Refer to the related man pages for the proper methods of setting these environment variables.

OPD requires some Perl modules to be installed before running. If they are not installed, you will receive a detailed error message listing which modules need to be installed. You have two options to install these modules:

1. Use a tool such as CPAN to download and install the required modules.
2. Launch the OSCAR installer (see Section 5.2) and then quit immediately when the GUI window appears. This is the preferred method, since the OSCAR installer will install of its own prerequisites (i.e., the process is automated).

Upon launching OPD, select a repository for packages from the choices listed. OPD will then connect to that repository and query it for a list of available packages. Select all the packages that you would like to download, and then use the "download" command to actually download them.

Use the "help" command in OPD for listings of additional commands and help.

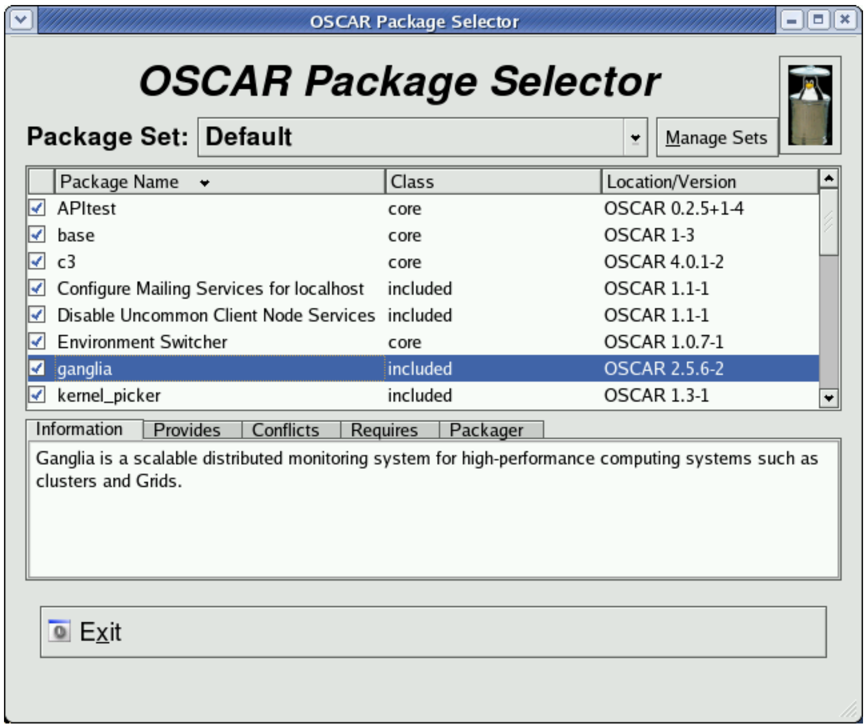
5.4 Selecting Packages to Install

Note: This step is optional.

If you wish to change the list of packages that are installed, click on the <**`Select OSCAR Packages To Install`**> button. This step is optional – by default all packages directly included in OSCAR are selected and installed. However, if you downloaded any additional packages, e.g., via OPD/OPDer, they will not be selected for installation by default. Therefore you will need to click this button and select the appropriate OSCAR Packages to install on the cluster.

When you click on the button, a window similar to the one shown in Figure 2 appears. Each of the packages that the OSCAR installer has found are listed in the main frame. Core packages must be installed and cannot be unselected. Included packages can be unselected if desired.

[6] Note, if using OPD directly, the packages must be downloaded before the <**`Select OSCAR Packages To Install`**> step (see Section 5.4) .



	Package Name	Class	Location/Version
✓	APItest	core	OSCAR 0.2.5+1-4
✓	base	core	OSCAR 1-3
✓	c3	core	OSCAR 4.0.1-2
✓	Configure Mailing Services for localhost	included	OSCAR 1.1-1
✓	Disable Uncommon Client Node Services	included	OSCAR 1.1-1
✓	Environment Switcher	core	OSCAR 1.0.7-1
✓	ganglia	included	OSCAR 2.5.6-2
✓	kernel_picker	included	OSCAR 1.3-1

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids.

Figure 2: OSCAR package selection.

Note that this window only shows *OSCAR packages* – it does not show individual RPMs. Once you have a selected a set of OSCAR packages to install, click on the <**`Exit`**> button to save your selections and return to the main OSCAR window. Note that closing the window yields the same result and there is no way of 'defaulting' to the original settings, so make sure your package list is complete before proceeding to the next step.

5.5 Configuring OSCAR Packages

Note: This step is optional.

Some OSCAR packages allow themselves to be configured. Clicking on the <**`Configure Selected OSCAR Packages`**> button will bring up a window listing all the packages that can be configured. Figure 3 shows a sample with only the Environment Switcher package listed.

Clicking on any of the packages' <**`Config`**> button will bring up a panel for configuring that package. Select whatever options are appropriate for that package, and then click on the <**`Save`**> button to save your selections, or the <**`Cancel`**> button to cancel all of your selections and leave the original settings. If you have saved your changes but want to go back to the default settings, simply click on the <**`Default Configuration`**> button and then the <**`Save`**> button to revert to the original settings.

This step is optional. If you do not click on the <**`Configure Selected OSCAR Packages`**> button, defaults for all packages will be used.

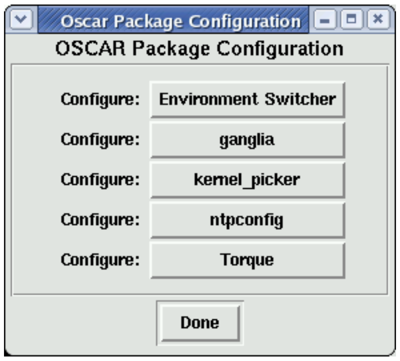

Oscar Package Configuration
OSCAR Package Configuration
Configure: Environment Switcher
Configure: ganglia
Configure: kernel_picker
Configure: ntpconfig
Configure: Torque
Done

Figure 3: OSCAR package configuration.

5.5.1 Selecting a Default MPI Implementation

Although multiple MPI implementations can be installed, only one can be "active" for each user at a time. Specifically, each user's path needs to be set to refer to a "default" MPI that will be used for all commands. The Environment Switcher package provides a convenient mechanism for switching between multiple MPI implementations. Section 6.13 contains more details about this package (page 38).

The Environment Switcher package is mentioned now, however, because its configuration panel allows you to select which MPI implementation will be the initial "default" for all users. OSCAR currently includes two MPI implementations: LAM/MPI and MPICH. Using Environment Switcher's configuration panel, you can select one of these two to be the cluster's default MPI.

You can change this default setting later – see Section 6.13 for more details.

When you close the main Configuration window, the following benign warning may appear in the shell window (it is safe to ignore):

```
Tag "mpi" does not seem to exist yet.  Skipping.
```

5.6 Install OSCAR Server Packages

Press the <**`Install OSCAR Server Packages`**> button. This will invoke the installation of various RPMs and auxiliary configuration on the server node. Execution may take several minutes; text output and status messages will appear in the shell window.

A popup will appear indicating the success or failure of this step. Click on the <**`Close`**> button to dismiss it.

5.7 Build OSCAR Client Image

Before pressing the <**`Build OSCAR Client Image`**>, ensure that the following conditions on the server are true:

- Ensure that the SSH daemon's configuration file (`/etc/ssh/sshd_config`) on the headnode has `PermitRootLogin` set to `yes`. After the OSCAR installation, you may set this back to `no` (if you want), but it needs to be `yes` during the install because the config file is copied to the client nodes, and `root` *must* be able to login to the client nodes remotely.

- By the same token, ensure that TCP wrappers settings are not "too tight". The `/etc/hosts.allow` and `/etc/hosts.deny` files should allow all traffic from the entire private subnet.

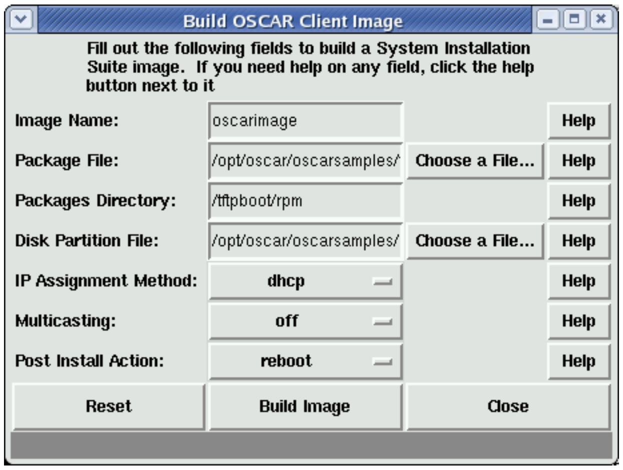
- Also, beware of firewall software that restricts traffic in the private subnet.

If these conditions are not met, the installation may fail during this step or later steps.

Press the <**`Build OSCAR Client Image`**> button. A dialog will be displayed. In most cases, the defaults will be sufficient. You should verify that the disk partition file is the proper type for your client nodes. The sample files have the disk type as the last part of the filename. You may also want to change the post installation action and the IP assignment methods. **It is important to note that if you wish to use automatic reboot, you should make sure the BIOS on each client is set to boot from the local hard drive before attempting a network boot by default. If you have to change the boot order to do a network boot before a disk boot to install your client machines, you should not use automatic reboot.**

Building the image may take several minutes; the green progress bar on the bottom of the window will indicate how far along the process is. There is a lot of output in the console window during the build. It is normal to see some warning messages in the console. You can safely ignore these messages and wait for the final popup window announcing the success or failure of the overall image build.

A sample dialog is shown in Figure 4.


Build OSCAR Client Image

Fill out the following fields to build a System Installation Suite image. If you need help on any field, click the help button next to it

Image Name: oscarimage
Package File: /opt/oscar/oscarsamples/
Packages Directory: /tftpboot/rpm
Disk Partition File: /opt/oscar/oscarsamples/
IP Assignment Method: dhcp
Multicasting: off
Post Install Action: reboot

Figure 4: Build the image.

Customizing your image. The defaults of this panel use the sample disk partition and RPM package files that can be found in the `oscarsamples` directory. You may want to customize these files to make the image suit your particular requirements.

Disk partitioning. The disk partition file contains a line for each partition desired, where each line is in the following format:

```
<partition> <size in megabytes> <type> <mount point> <options>
```

Here is a sample (for a SCSI disk):

```
/dev/sda1        24   ext3  /boot      defaults
/dev/sda5        128  swap
/dev/sda6        *    ext3  /          defaults
nfs_oscar:/home  -    nfs   /home      rw
```

An * in the size column causes that partition to grow to fill the entire disk. You can create your own partition files, but make sure that you do not exceed the physical capacity of your client hardware. Also be careful to not specify duplicate filesystems as this will cause problems during the installation. The sample listed above, and some others, are in the `oscarsamples` directory.

Package lists. The package list is simply a list of RPM file names (one per line). Be sure to include all prerequisites that any packages you might add. You do not need to specify the version, architecture, or extension of the RPM filename. For example, `bash-2.05-8.i386.rpm` need only be listed as "`bash`".

Custom kernels. If you want to use a customized kernel, you can add it to the image after it is built (after installing the server OSCAR packages, but before building the client image). See the `kernel_picker` application description in Section 6.4 on page 34.

Build the Image. Once you are satisfied with the input, click the <**`Build Image`**> button. When the image completes, a popup window will appear indicating whether the build succeeded or failed. If successful, click the <**`Close`**> button to close the popup, and then press the <**`Close`**> button on the build image window. You will be back at the main OSCAR wizard menu.

If the build fails, look through the console output[7] for some indication as to what happened to cause the failure. Common causes include: prerequisite failure, ran out of disk space, and missing package files. Also see the Release Notes for this version of OSCAR in Section 3 (page 8).

5.8 Define OSCAR Clients

Press the <**`Define OSCAR Clients`**> button. In the dialog box that is displayed, enter the appropriate information. Although the defaults will be sufficient for most cases, you will need to enter a value in the *`Number of Hosts`* field to specify how many clients you want to create.

1. The *`Image Name`* field should specify the image name that was used to create the image in the previous step.

2. The *`Domain Name`* field should be used to specify the client's IP domain name. It should contain the server node's domain (if it has one); if the server does not have a domain name, the default name `oscardomain` will be put in the field (although you may change it). **This field *must* have a value – it cannot be blank.** Note that especially for compute nodes on a private network, the domain name does not necessarily matter much. The domain name supplied in this field is used to form the fully-qualified name of each host in the OSCAR cluster. For example: `oscarnode1.oscardomain`, `oscarnode2.oscardomain`, etc. If your compute nodes are on a public network, you may want to use the "real" domain name that is part of their fully-qualified domain names.

3. The *`Base name`* field is used to specify the first part of the client name and hostname. It will have an index appended to the end of it. **This name *cannot* contain an underscore character "_" or a period ".".**

4. The *`Number of Hosts`* field specifies how many clients to create. **This number must be greater than 0.**

[7]Note that all console output is also spooled into the file `oscarinstall.log`.

5. The `Starting Number` specifies the index to append to the `Base Name` to derive the first client name. It will be incremented for each subsequent client.

6. The `Padding` specifies the number of digits to pad the client names, e.g., 3 digits would yield oscarnode001. The default is 0 to have no padding between base name and number (index).

7. The `Starting IP` specifies the IP address of the first client. It will be incremented for each subsequent client. See Footnote 3 on page 18 for more information on how to pick a starting IP address.

   Clients will be given IP addresses starting with this IP address, and incrementing by 1 for each successive client. Ensure that the range of $[starting\_ip, (starting\_ip + num\_clients)]$ does not conflict with the IP addresses of any other nodes on your network.

   IMPORTANT NOTE: Be sure that the resulting range of IP addresses does *not* include typical broadcast addresses such as $X.Y.Z.255$! If you have more hosts than will fit in a single address range, see the note at the end of this section about how to make multiple IP address ranges.

8. The `Subnet Mask` specifies the IP netmask for all clients. See Footnote 4 on page 18 for more information on how to select a netmask for your cluster.

9. The `Default Gateway` specifies the default route for all clients.

When finished entering information, press the <**`Addclients`**> button. When those clients have been created in the database, a popup will appear indicating the completion status. A sample dialog is shown in Figure 5.

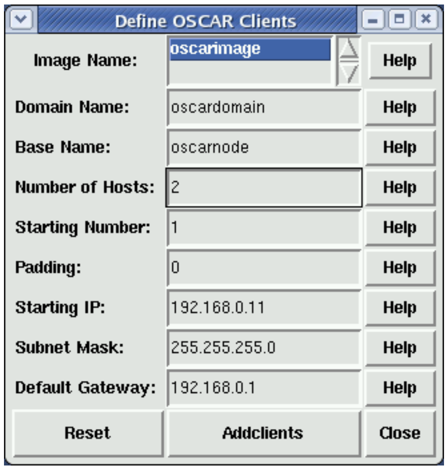


Image Name:	oscarimage
Domain Name:	oscardomain
Base Name:	oscarnode
Number of Hosts:	2
Starting Number:	1
Padding:	0
Starting IP:	192.168.0.11
Subnet Mask:	255.255.255.0
Default Gateway:	192.168.0.1

Figure 5: Define the Clients.

Note that this step can be executed multiple times. The GUI panel that is presented has limited flexibility in IP address numbering – the starting IP address will only increment the least significant byte by one for each successive client. Hence, if you need to define more than 254 clients (beware of broadcast addresses!), you will need to run this step multiple times and change the starting IP address. There is no need to close the panel and return to the main OSCAR menu before executing it again; simply edit the information and click on the <**`Addclients`**> button as many times as is required.

Additionally, you can run this step multiple times to use more structured IP addressing schemes. With a larger cluster, for example, it may be desirable to assign IP addresses based on the top-level switch that they are connected to. For example, the 32 clients connected to switch 1 should have an address of the form 192.168.1.x. The next 32 clients will be connected to switch 2, and should therefore have an address of the form 192.168.2.x. And so on.

After all clients have been created, you may press the <**`Close`**> button in the build clients dialogue and continue with the next step.

5.9 Setup Networking

The MAC address of a client is a twelve hex-digit hardware address embedded in the client's ethernet adapter. For example, "`00:0A:CC:01:02:03`", as opposed to the familiar format of IP addresses. These MAC addresses uniquely identify client machines on a network before they are assigned IP addresses. DHCP uses the MAC address to assign IP addresses to the clients.

In order to collect the MAC addresses, press the <**`Setup Networking`**> button. The OSCAR network utility dialog box will be displayed. To use this tool, you will need to know how to network boot your client nodes, or have a file that lists all the MACs from your cluster. For instructions on doing network booting, see Appendix A. A sample dialog is shown in Figure 6.

If you need to collect the MACs in your cluster, start the collection by pressing the <**`Collect MAC Address`**> button and then network boot the first client. As the clients boot up, their MAC addresses will show up in the left hand window. You have multiple options for assigning MACs to nodes; you can either:

- manually select MAC address and the appropriate client in the right side window. Click <**`Assign MAC to Node`**> to associate that MAC address with that node.
- click <**`Assign all MACs`**> button to assign all the MACs in the left hand window to all the open nodes in the right hand window.

Some notes that are relevant to collecting MAC addresses from the network:

- The <**`Dynamic DHCP Update`**> checkbox at the bottom right of the window controls refreshing the DHCP server. If it is selected (the default), the DHCP server configuration will be refreshed each time a MAC is assigned to a node. Note that if the DHCP reconfiguration takes place quick enough, you may not need to reboot the nodes a second time (i.e., if the DHCP server answers the request quick enough, the node may start downloading its image immediately).

  If this option is off, you will need to click the <**`Configure DHCP Server`**> (at least once) to give it the associations between MACs and IP addresses.
- To remove extraneous MAC addresses from the left hand window (e.g., if the collector finds MACs that are not part of your cluster), select the address and click on the <**`Remove`**> button. Or click on the <**`Remove All`**> button to remove all of them.
- At any time, you may click on the <**`Export MACs to file...`**> button to save the MAC address list to a file. If you need to re-run the OSCAR installation, you can later click on <**`Import MACs from file...`**> to import this file rather than re-collecting all the MACs. *Currently, MACs can only be saved in /root which is the default directory.*
- When you have collected all of the MAC addresses, click the <**`Stop Collecting MACs`**> button. If you do not have <**`Dynamic DHCP update`**> selected, you need to click the <**`Configure DHCP Server`**> button to configure the DHCP server.

- You *must* click on the <**`Stop Collecting MACs`**> before closing the MAC Address Collection window!

This menu also allows you to choose your remote boot method. You should do one of two things after collecting your MAC addresses but before leaving the menu:

- The <**`Build Autoinstall Floppy`**> button will build a boot floppy for client nodes that do not support PXE booting.
- The <**`Setup Network Boot`**> button will configure the server to answer PXE boot requests if your client hardware supports it. See Appendix A for more details.

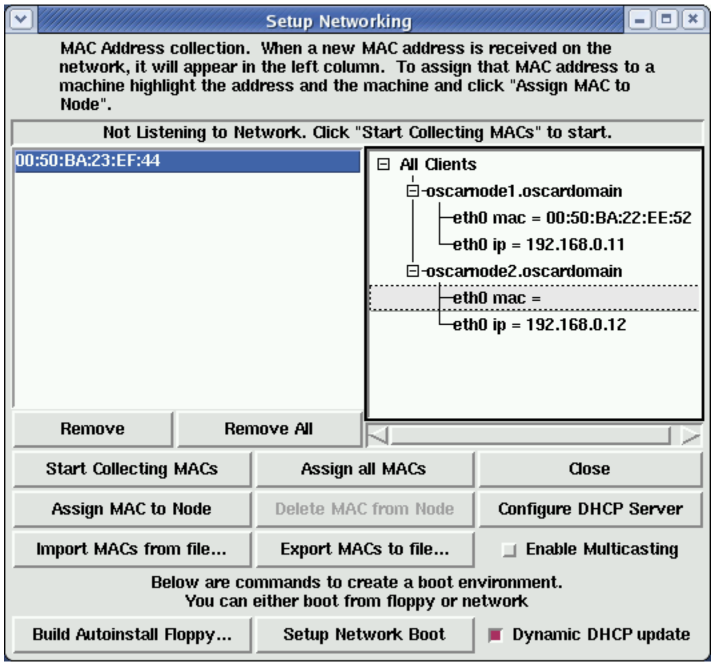
If your network switch supports multicasting, there is a new feature in OSCAR 3.0 which uses multicast to push files to the clients. To enable this feature simply click on the <**`Enable Multicasting`**> checkbox.

Once this feature is enabled, rsync will not be used for file distribution but instead by a program called Flamethrower which is bundled with SystemImager.

Since this new feature is still in its early stage, we recommend only the adventurous to try it as it may not work on all networking gear. Some have reported favorable results with the Force10 and HP 4000M network switches.

In case you cannot get this feature working and want to switch back to using rsync, simply go back to the <**`Setup Networking`**> menu, make sure that the <**`Enable Multicasting`**> checkbox is disabled (should be disabled by default) and then click on <**`Configure DHCP Server`**> - this should revert back to the default settings.

When you have collected the addresses for all your client nodes and completed the networks setup, press the <**`Close`**> button.


Setup Networking

MAC Address collection. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click "Assign MAC to Node".

Not Listening to Network. Click "Start Collecting MACs" to start.

```
00:50:BA:23:EF:44
```

```
All Clients
  oscarnode1.oscardomain
    eth0 mac = 00:50:BA:22:EE:52
    eth0 ip = 192.168.0.11
  oscarnode2.oscardomain
    eth0 mac =
    eth0 ip = 192.168.0.12
```

Below are commands to create a boot environment.
You can either boot from floppy or network

Enable Multicasting: ☐
Dynamic DHCP update: ■

Figure 6: Collect client MAC addresses.

5.10 Client Installations

During this phase, you will network boot your client nodes and they will automatically be installed and configured. For a detailed explanation of what happens during client installation, see Appendix B.

5.10.1 Network boot the client nodes

See Appendix A for instructions on network booting clients.

Network boot all of your clients. As each machine boots, it will automatically start downloading and installing the OSCAR image from the server node.

5.10.2 Check completion status of nodes

After several minutes, the clients should complete the installation. You can watch the client consoles to monitor the progress. Depending on the Post Installation Action you selected when building the image, the clients will either halt, reboot, or beep incessantly when the installation is completed.

The time required for installation depends on the capabilities of your server, your clients, your network, and the number of simultaneous client installations. Generally, it should complete within several minutes.

5.10.3 Reboot the client nodes

After confirming that a client has completed its installation, you should reboot the node from its hard drive. If you chose to have your clients reboot after installation, they will do this on their own. If the clients are not set to reboot, you must manually reboot them. The filesystems will have been unmounted so it is safe to simply reset or power cycle them.

Note: If you had to change the BIOS boot order on the client to do a network boot before booting from the local disk, you will need to reset the order to prevent the node from trying to do another network install.

5.11 Complete the Cluster Setup

Ensure that all client nodes have fully booted before proceeding with this step.

Press the <**`Complete Cluster Setup`**> button. This will run the final installation configurations scripts from each OSCAR software package, and perform various cleanup and re-initialization functions. This step can be repeated should networking problems or other types of errors prevent it from being successful the first time.

A popup window will indicate the success or failure of this step. Press the <**`Close`**> button to dismiss it.

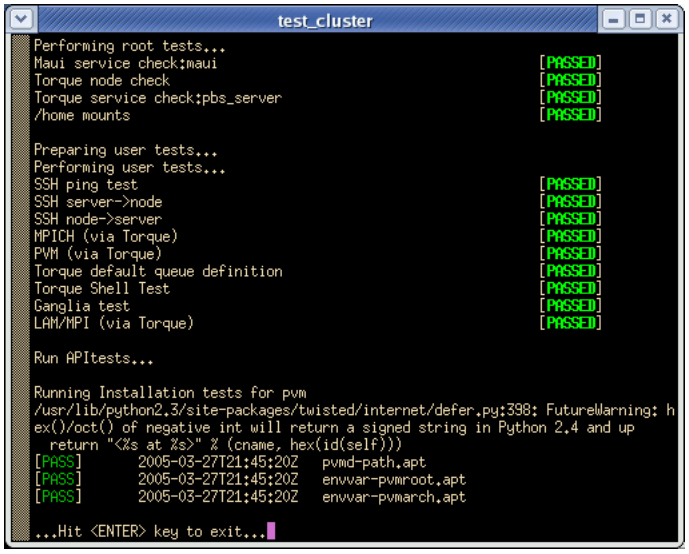
5.12 Test Cluster Setup

A simplistic test suite is provided in OSCAR to ensure that the key cluster components (OpenSSH, Torque, MPI, PVM, etc.) are functioning properly.

Press the <**`Test Cluster Setup`**> button. This will open a separate window to run the tests in. The cluster's basic services are checked and then a set of `root` and user level tests are run.

A sample dialog is shown in Figure 7. If any of the test fail, then there may be problem with your installation.

At the beginning of the test, when the Torque server is shut down, you may see a benign error message about the `pbsnodes` command not being able to connect to the Torque server. This is safe to ignore.



```
Performing root tests...
Maui service check:maui                                                       [PASSED]
Torque node check                                                             [PASSED]
Torque service check:pbs_server                                               [PASSED]
/home mounts                                                                  [PASSED]

Preparing user tests...
Performing user tests...
SSH ping test                                                                 [PASSED]
SSH server->node                                                              [PASSED]
SSH node->server                                                              [PASSED]
MPICH (via Torque)                                                            [PASSED]
PVM (via Torque)                                                              [PASSED]
Torque default queue definition                                               [PASSED]
Torque Shell Test                                                             [PASSED]
Ganglia test                                                                  [PASSED]
LAM/MPI (via Torque)                                                          [PASSED]

Run APItests...

Running Installation tests for pvm
/usr/lib/python2.3/site-packages/twisted/internet/defer.py:398: FutureWarning: h
ex()/oct() of negative int will return a signed string in Python 2.4 and up
  return "<%s at %s>" % (cname, hex(id(self)))
[PASS]       2005-03-27T21:45:20Z   pvmd-path.apt
[PASS]       2005-03-27T21:45:20Z   envvar-pvmroot.apt
[PASS]       2005-03-27T21:45:20Z   envvar-pvmarch.apt

...Hit <ENTER> key to exit...
```

Figure 7: Setup cluster tests

5.13 Congratulations!

Your cluster setup is now complete. Your cluster nodes should be ready for work.

Be sure to read Section 6 (starting on page 33) – it contains vital system administrator-level information on several of the individual packages that were installed as part of OSCAR.

5.14 Adding and Deleting client nodes

This section describes the steps need when it becomes necessary to add or delete client nodes. If you have already built your cluster successfully and would like to add or delete a client node, execute the following from the top-level OSCAR directory:

```
# ./install_cluster <device>
```

Like before, you must substitute the device name (e.g., `eth1`) for the server node's internal ethernet adapter in the above command. See Section 5.2 (page 19). Once the OSCAR wizard appears, you are ready to add or delete clients. Note that these steps will reuse the existing images made with the initial install, however, it will extend or contract the set of defined clients in the cluster.

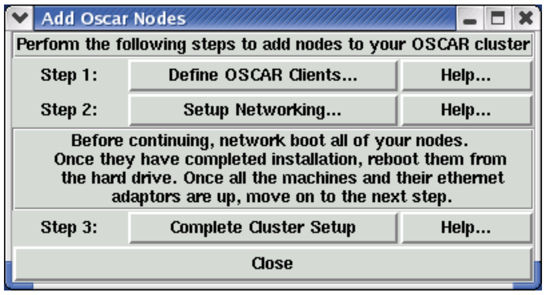
5.14.1 Adding OSCAR clients

Press the button of the wizard entitled <**`Add OSCAR Clients`**>. A sample dialog is shown in Figure 8. These steps should seem familiar – they are same as the initial install steps. Refer to Sections 5.8, 5.9, and 5.11.

Note that when adding nodes, in the Defining OSCAR Clients step, you will typically need to change the following fields to suit your particular configuration:

- Number of hosts
- Starting number

- Starting IP



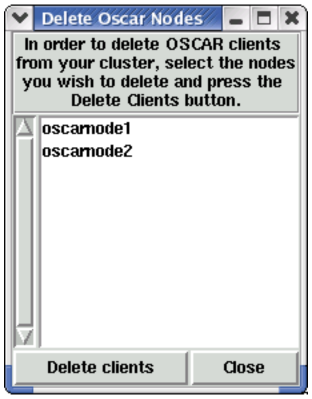
Perform the following steps to add nodes to your OSCAR cluster

Before continuing, network boot all of your nodes. Once they have completed installation, reboot them from the hard drive. Once all the machines and their ethernet adaptors are up, move on to the next step.

Figure 8: Adding OSCAR clients.

5.14.2 Deleting clients

Press the button of the wizard entitled <`Delete OSCAR Clients`>. A sample dialog is shown in Figure 9. Select the node(s) that you wish to delete and press the button <`Delete clients`>, then press <`Close`>.



In order to delete OSCAR clients from your cluster, select the nodes you wish to delete and press the Delete Clients button.

oscarnode1
oscarnode2

Figure 9: Deleting OSCAR clients.

5.15 Install/Uninstall OSCAR Packages

The package installation system is designed to help simplify adding and removing OSCAR packages after the initial system installation. The system is fairly straightforward for the user as packages are selected for the appropriate action from the wizard.

5.15.1 Selecting the Right Package

Because of the way OSCAR handles packages, two packages of the same name cannot currently coexist nicely on the system. Packages can be in one of two spots:

1. `$OSCAR_HOME/packages` – OSCAR installation directory

2. `/var/lib/oscar/package` – OPD download area

If there are multiple packages on the system of the same name, the package located in the OPD download area is the package that the system recognizes. It is possible that a package placed in /var/lib/oscar/package via a download or manually could get loaded into the database, while the original package that is located in `$OSCAR_HOME/packages` is the one that is actually installed. That is why this system does not support upgrades. That is planned for a future OSCAR release. This can be avoided by uninstalling the original package first before you download anything and then re-running the wizard. When the wizard first runs (`./install_cluster ethX`), the XML files are re-read for all packages and the database re-initialized.

5.15.2 Single Image Restriction

Currently, if more than one image is detected on your system, the package install/uninstall system will not run. Packages are installed and uninstalled to the clients (compute nodes), the server, and an image (singular). This release does not support the mapping of nodes to an image for package install/uninstall through the wizard. This single image restriction is instituted to help reduce the risk of error with this initial release, i.e., ”keep it simpler”. The system looks for images in `/var/lib/systemimager/images` and no other place.

5.15.3 Failures to Install and Uninstall

A package is not installed or uninstalled until it successfully installs or uninstalls on the compute nodes, the server, and a single image. We will run through an example problem as the best way of describing how to debug the system. As a note, the system does sanity checking to try to avoid the following situation.

For example, during the install process the clients may have been successfully installed, but the server installation died half way through for some unknown reason, and the install program exits with an error. If this happens, a second attempt to install will probably fail even if the server's problem is resolved.

The reason for this is that something happened on the clients and it succeeded, probably RPM's were installed. So the second time you go to install the package, the `rpm -Uvh` command on the nodes will fail.

So, what do you do?

Really, you have two options. The first is to look back through the log to see what commands were run, and undo those commands by hand. The second would be to manually push out the uninstall scripts to the compute nodes, and run them outside of the wizard environment. All scripts are re-runnable in OSCAR, and for serveral good reasons.

These options are not good ones, and we recognize this fact. We are working on better tools for general use. Generally, it is a difficult problem to judge error codes on remote systems and try to guess the appropriate actions to take. We have developed some additional software to start solving this problem, but there is still work that needs to be done in this area.

On the uninstall side of things, the picture is much simpler. The only action taken by the system is to run the package's uninstall script and judge the return code. So an uninstall failure is probably a faulty uninstall script.

Note if you hit the <**`Cancel`**> button or close down the window, some unexpected results may occur (post installation scripts will still run) - this is perfectly normal - please refer to the release notes in Section 3 for more info.

5.15.4 More Debugging Info

Some amount of trouble was taken to insure that decent debugging output was printed to the log – making best use of this output is your best bet in an error case.

5.16 Starting over – installing OSCAR again

If you feel that you want to start the cluster installation process over from scratch in order to recover from irresolvable errors, you can do so with the `start_over` script located in the `scripts` subdirectory.

It is important to note that `start_over` is *not* an uninstaller. That is, `start_over` does *not* guarantee to return the head node to the state that it was in before OSCAR was installed. It does a "best attempt" to do so, but the only guarantee that it provides is that the head node will be suitable for OSCAR re-installation. For example, the RedHat 7.x series ships with a LAM/MPI RPM. The OSCAR install process removes this RedHat-default RPM and installs a custom OSCAR-ized LAM/MPI RPM. The `start_over` script only removes the OSCAR-ized LAM/MPI RPM – it does not re-install the RedHat-default LAM/MPI RPM.

Another important fact to note is that because of the environment manipulation that was performed via `switcher` from the previous OSCAR install, *it is necessary to re-install OSCAR from a shell that was not tainted by the previous OSCAR installation*. Specifically, the `start_over` script can remove most files and packages that were installed by OSCAR, but it cannot chase down and patch up any currently-running user environments that were tainted by the OSCAR environment manipulation packages.

Ensuring to have an untainted environment can be done in one of two ways:

1. After running `start_over`, completely logout and log back in again before re-installing. *Simply launching a new shell may not be sufficient* (e.g., if the parent environment was tainted by the previous OSCAR install). This will completely erase the previous OSCAR installation's effect on the environment in all user shells, and establish a set of new, untainted user environments.

2. Use a shell that was established *before* the previous OSCAR installation was established. Although perhaps not entirely intuitive, this may include the shell was that initially used to install the previous OSCAR installation.

Note that the logout/login method is *strongly* encouraged, as it may be difficult to otherwise absolutely guarantee that a given shell/window has an untainted environment.

6 Package-Specific Installation Notes

The following sections provide package-specific notes regarding installation.

6.1 APItest

The *APItest* tool has been developed as part of the SciDAC: Scalable System Software (SSS) project to provide a general testing framework. Tests are written in a simple XML format which can include either commands or embedded scripts (e.g., `/bin/sh`, `/usr/bin/perl`, etc.). The framework allows for success/failure tests to be determined by literal or regular expression matches on the output or by checking return code status.

These basic tests are grouped together in batches to form more elaborate tests. All tests (including test-batches) can express dependencies and the framework orders them for proper execution. The results can be viewed from a basic command-line interface or a graphical web interface.

For further details see the supplied documentation and the SSS electronic notebooks at `http://www.scidac.org/ScalableSystems`.

6.2 Disabling Services

The `disable-services` OSCAR package disables the following services (if they exist) on the client nodes:

- Incoming mail service: the `sendmail`, `exim`, and `postfix` daemons are disabled. This prevents incoming mail from being received on the nodes. Note that outgoing mail is not disabled; most outgoing mail is sent immediately. If there is some transient failure and the mail is not sent immediately, it will go to the mail service's queue. A `cron.hourly` crontab is installed that runs every hour to trigger the queue in case this happens.
- Kudzu: the Kudzu service looks for new hardware, usually upon boot up. At boot time, this process takes over 30 seconds. It is disabled in order to speed up the booting of individual nodes.
- `slocate`: the `slocate` service runs a top-level `find` command over all local filesystems periodically (some Linux distributions have it set to run daily, others have it set to run weekly) in order to index all filenames for quick lookup using the `locate` command. The top-level `find` command takes significant amounts of system resources to run, and is therefore disabled.
- `makewhatis`: the `makewhatis` command is run via crontab (sometimes daily, sometimes weekly – depending on the specific Linux distribution) to generate manual page indexes. As with `slocate`, this command takes significant amounts of system resources to run, and is therefore disabled.

Note that these services are not uninstalled – they are simply disabled. Administrators are free to re-enable them if they wish.

6.3 Ganglia

The following is an overview about Ganglia which is taken from the official website at `http://www.ganglia.info`:

"Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids. It is based on a hierarchical design targeted at federations of clusters. It leverages widely used technologies such as XML for data representation, XDR for compact, portable data transport, and RRDtool for data storage and visualization. It uses carefully engineered data structures and algorithms to achieve very low per-node overheads and high concurrency. The implementation is robust, has been ported to an extensive set of operating systems and processor architectures, and is currently in use on over 500 clusters around the world. It has been used to link clusters across university campuses and around the world and can scale to handle clusters with 2000 nodes."

Ganglia is an included package in OSCAR and it is highly recommended that you select it to be installed as it is a great tool for monitoring and visualizing usage of your cluster.

By default, it is assumed that `eth0` is the network interface your compute nodes use to communicate with the headnode. If this is not the case, then you will need to configure the package prior to installation and choose the correct interface. Other than that, minimal configuration is required as the other configurable options are mostly to do with the names of the cluster, grid and owner.

6.4 Selecting a Different Kernel with `kernel_picker`

`kernel_picker` is a Perl script which allows a user to install a given kernel into an OSCAR image different from the one which is installed by default. After step 1, but before step 2, you can run `kernel_picker` to substitute a given kernel into your OSCAR (SIS) image. If executed with no command line options, you will be prompted for all information. If you use any command line options, the program will assume that you know what you are doing and prompt you *only* for information which is required for correct execution.

The `kernel_picker` program assumes that the optional OSCAR image files you wish to use reside in a subdirectory in the `/var/lib/systemimager/images` directory. By default, the original OSCAR image is in a subdirectory named `oscarimage`.

`kernel_picker` is installed in the `/opt/kernel_picker/bin` directory. Documentation is available in HTML, PostScript, PDF, plain text, and manpage formats. To see the manpage documentation, type the following at a Unix command prompt:

```
  $ man /opt/kernel_picker/man/man1/kernel_picker.1
```

6.5 Uninstalling LAM/MPI

Please note that there is a known bug in the `lam-switcher-modulefile` RPM such that if you use the OSCAR infrastructure to remove the LAM/MPI package and LAM is set as OSCAR's default MPI implementation, you will see errors from the switcher package. You can clean up these problems by running (as root):

```
# switcher mpi = <some_other_mpi_implementation>
```

Or explicitly state that there is no default OSCAR MPI implementation:

```
# switcher mpi = none
```

6.6 Maui Scheduler

Note that all the URLs in the Maui configuration file (`/opt/maui/maui.cfg`) have since changed since the Maui RPM was created. Look for all Maui scheduler-related URLs under http://www.clusterresources.com/.

6.7 Networking Package

The networking package was created to set up the OSCAR server as a caching nameserver.

By default, this package will set up a caching nameserver on the OSCAR server.

6.8 Managing NTP for the OSCAR Server and Clients

NTP is the Network Time Protocol which is used to synchronize the computer clock to external sources of time. The `ntpd` daemon can operate as a client (by connecting to other NTP servers to get the current time) and as a server (by providing the current time to other NTP clients).

The ntpconfig package which configures NTP for your OSCAR cluster, has been completely re-written since OSCAR 4.1.

6.8.1 Configuring ntpconfig

If not configured, the package will set up your cluster nodes such that they will use the headnode as the time server and sync time against it. This mode is useful if you do not have an Internet connection.

The package will not overwrite any of your existing configurations for NTP, e.g. during the initial OS installation on your headnode, you set up NTP to sync time with `time.apple.com`, the package will not change that and will simply set up your headnode as the time server for your cluster nodes. As a result, your compute nodes' time will be synced relative to `time.apple.com`'s clock.

If you choose to configure the ntpconfig package, you can select a NTP server to sync time with (we recommend `pool.ntp.org`), and your headnode will then synchronize clock with the NTP server you chose and any other time servers previously configured manually by modifying the `/etc/ntp.conf` file.

If you have previously configured NTP using the ntpconfig package, and have configured it to use another server, the previous setting will be overwritten.

6.8.2 Enabling/Disabling the NTP Service

By default, the `ntpd` daemon is configured to start at boot time in run levels 2 through 5. If for some reason you want to disable NTP without actually uninstalling it, execute the following commands:

```
# /etc/init.d/ntpd stop
# /sbin/chkconfig --level 2345 ntpd off
```

This will not only stop any currently running `ntpd` daemon, but also prevent NTP from starting up at boot time.

NOTE: You must be `root` to execute these commands.

To restart NTP and make NTP start up at boot time, execute the following commands:

```
# /etc/init.d/ntpd restart
# /sbin/chkconfig --level 2345 ntpd on
```

For more information on NTP, see the (rather lengthy) documentation at `http://www.ntp.org/`.

6.9 The OSCAR Password Installer and User Management (OPIUM)

Currently, OPIUM manages users and passwords by replicating files on a triggered basis using the sync_files utility. The user account files are copied to each node on a regular interval. It is possible to add your own files to this list, by editing the configuation file `/opt/sync_files/etc/sync_files.conf`.

6.9.1 SSH Configuration

The OPIUM package also handles the SSH key setup. This is done by placing startup scripts in `/etc/profile.d` which generate SSH user key files if they don't already exist. It also generates `authorized_keys` files which enable users to traverse the cluster securely without entering passwords. Since these files reside in the users' home directories, it is necessary for the home directory to be mounted on a node in order for a user to `ssh` to it. Also, a user must log on to the head node in order for the keys to be generated, as the `/etc/profile.d` scripts are not installed on the compute nodes in OSCAR 4.1. SSH1 and SSH2 key files are supported.

6.10 Packet Filtering with pfilter

pfilter is a firewall compiler. The **pfilter** package is used to control the packet filtering capabilities available in the Linux kernel. It takes in high-level firewall directives, and produces a complete firewall output commands file that can be turned on or off like other Linux services. Like other compilers, **pfilter** adds appropriate "glue" code to the compiled output. **pfilter**'s added "glue" code consists of common things that are done by any good firewall, including turning on TCP networking protective features.

When OSCAR was installed, it merged any pre-existing server **pfilter** installation configuration into a new configuration, and created new client machine **pfilter** configurations. The resulting **pfilter** configurations do the following:

- any network connections that were specifically allowed in the former server **pfilter** installation configuration are still allowed to the main OSCAR server node
- the main OSCAR server node and all client OSCAR nodes allow`ssh` logins from anywhere
- the main OSCAR server node has http access enabled from anywhere
- any remaining network connections from outside the cluster are blocked
- the logging of bad network packets to syslog is turned off
- all network communication of any kind between nodes in the cluster is enabled

If the main OSCAR server has two or more network interfaces, **pfilter** will attempt to determine which network interface is the public interface, and then enable packet forwarding and network address translation for the remaining interfaces if it can.

pfilter is turned on by default – this is considered good "defense in depth" security for a cluster (see Section D for more information on cluster security). If for some reason you need to disable **pfilter** packet filtering (perhaps for debugging – disabling it permanently is not recomended), you can disable **pfilter** on subsequent system boots with the following command:

```
# chkconfig --level=2345 pfilter off
```

To turn on **pfilter** packet filtering immediately, execute the following command:

```
# service pfilter start
```

For more information on how to use and configure **pfilter**, see the `pfilter(8)` and `pfilter.conf(5)` and `pfilter.rulesets(5)` man pages.

6.11 PVM APItests: Installation

The PVM package includes several installation tests written in an XML based format for use with the *APItest* testing framework. These tests check for installation and configuration setting for the OSCAR installed PVM package. See also the *APItest* package for more details.

6.12 Managing machines and images in SIS

During the life of your cluster, you may want to delete unused images, create new images, or change the image that a client uses. Currently OSCAR doesn't have a direct interface to do this, but you can use the SIS commands directly. Here are some useful examples:

- To list all defined machines, run:

  ```
  mksimachine --List
  ```

- To list all defined images, run:

  ```
  mksiimage --List
  ```

- To delete an image, run:

  ```
  mksiimage --Delete --name <imagename>
  ```

- To change which image a machine will install, run:

  ```
  mksimachine --Update --name <machinename> --image <imagename>
  ```

There is also a SIS GUI that is availble. Start it by running `tksis`.
More details on these commands can be obtained from their respective man pages.

6.12.1 Multicast Installs with SystemImager

SIS now includes multicast install capability with SystemImager v3.2.x and Flamethrower v1.0.x. Working with multicast can prove very beneficial, especially for large sites. However, multicast can be a tricky thing to get working reliably, based on networking equipment, multicast tuning parameters, machine speed, etc. If you are interested in giving multicast a try, see the ”HOWTO Use Flamethrower for Multicast Installs” section in the SystemImager manual (http://www.systemimager.org/documentation/), and please provide us with feedback on your success!

6.13 Which MPI do you want to use?

OSCAR has a generalized mechanism to both set a system-level default MPI implementation, and also to allow users to override the system-level default with their own choice of MPI implementation.

This allows multiple MPI implementations to be installed on an OSCAR cluster (e.g., LAM/MPI and MPICH), yet still provide unambiguous MPI implementation selection for each user such that "`mpicc foo.c -o foo`" will give deterministic results.

6.13.1 Setting the system-level default

The system-level default MPI implementation can be set in two different (yet equivalent) ways:

1. During the OSCAR installation, the GUI will prompt asking which MPI should be the system-level default. This will set the default for all users on the system who do not provide their own individual MPI settings.

2. As `root`, execute the command:

   ```
   # switcher mpi --list
   ```

   This will list all the MPI implementations available. To set the system-level default, execute the command:

   ```
   # switcher mpi = name --system
   ```

   where "name" is one of the names from the output of the `--list` command.

NOTE: System-level defaults for `switcher` are currently propogated to the nodes on a periodic basis. If you set the system-level MPI default, you will either need to wait until the next automatic "push" of configuration information, or manually execute the `/opt/sync_files/bin/sync_files` command to push the changes to the compute nodes.

NOTE: Using the `switcher` command to change the default MPI implementation will modify the `PATH` and `MANPATH` for all *future* shell invocations – it does *not* change the environment of the shell in which it was invoked. For example:

```
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher mpi = mpich-4.5.6 --system
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# bash
# which mpicc
/opt/mpich-4.5.6/bin/mpicc
```

If you wish to have your current shell reflect the status of your switcher settings, you must run the "`switcher-reload`" command. For example:

```
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher mpi = mpich-4.5.6 --system
# which mpicc
/opt/lam-1.2.3/bin/mpicc
# switcher-reload
# which mpicc
/opt/mpich-4.5.6/bin/mpicc
```

Note that this is *only* necessary if you want to change your current environment. All new shells (including scripts) will automatically get the new switcher settings.

6.13.2 Setting the user-level default

Setting a user-level default is essentially the same as setting the system-level default, except without the `--system` argument. This will set the user-level default instead of the system-level default:

```
$ switcher mpi = lam-1.2.3
```

Using the special name `none` will indicate that no module should be loaded for the `mpi` tag. It is most often used by users to specify that they do not want a particular software package loaded.

```
$ switcher mpi = none
```

Removing a user default (and therefore reverting to the system-level default) is done by removing the `default` attribute:

```
$ switcher mpi --show
user:default=mpich-1.2.4
system:exists=true
$ switcher mpi --rm-attr default
$ switcher mpi --show
system:default=lam-6.5.6
system:exists=true
```

6.13.3 Use `switcher` with care!

`switcher` immediately affects the environment of all future shell invocations (including the environment of scripts). To get a full list of options available, read the `switcher(1)` man page, and/or run `switcher --help` (or `switcher --more-help`).

6.14 Torque Resource Manager and Maui Scheduler

Torque serves as the job launcher and batch queueing system for OSCAR. Torque includes a basic FIFO scheduler with it, but is disabled by default in OSCAR. A more robust, open source scheduler named "Maui" is used instead.

Basic Torque functionality is tested by OSCAR's test suite, and is also used to launch jobs when testing other software included in OSCAR. If the Torque test passes, Torque and Maui are up and working.

If the users of your OSCAR cluster have not used Torque before, you can expect somewhat of a learning curve. The OSCAR user's documentation contains some useful information to get them started. The user's document instructs users to ask the system administrator to provide them with sample Torque scripts used in the OSCAR test suite. Once the OSCAR test suite has been run (step 6 in the install process), these scripts can be found in the home directory of the `oscartst` user if the tests have been run previously.

6.14.1 Configuring Torque

By default, Torque installs without any queues or cluster specific paremeters defined. OSCAR configures Torque with sensible defaults based on what it finds in the SIS database. When the "Complete Cluster Setup" step is executed from the wizard, the `post_install` script from the Torque package in OSCAR is called. The `post_install` configures only Torque parameters that are non-existent, so as not to overwrite local customizations by the system administrator. However, if you wish to force the default values back in place over any local customizations, the `post_install` script can be invoked manually with a `--default` option. This will revert all values to the original OSCAR settings.

`qmgr` can be used to configure queues and Torque server parameters. The OSCAR Torque `post_install` script (located off the top-level OSCAR installation directory in `packages/torque/scripts`) uses `qmgr` behind the scenes. There are man pages available, but reading the Torque admin guide is the best way to learn how to use it. It is available on Torque's homepage, listed below. You will have to create an account on their site in order to download the admin guide.

6.14.2 Torque Resources

Arbitrary node properties can be set by the administrator. Torque calls these properties "resources". These resources can be specified on the `qsub` command line when a user submits a job. This allows a user to restrict their jobs to run only on nodes exhibiting certain properties. If some nodes of a cluster have more memory, a different network, faster prcoessors, etc., jobs can be submitted so they only run a specific subset. These properties are stored in plain text in `/usr/spool/PBS/server_priv/nodes`. However, if adjusted in the plain text file, the Torque server must be restarted in order for changes to take effect. The more elaborate method is to use the `qmgr` command to modify node properties via the Torque API. OSCAR gives each node a starting property of "`all`".

6.14.3 An FAQueue

A popular misconception about Torque queues is that they are bound to a group of nodes. This is false. If you have a four node queue defined, it is not associated with any specific nodes. You can think of a queue as a multidimensional box that a job must fit in in order to allow submission. That is, the submitted parameters must fit within certain max and min values for nodes, ppn (procs per node), walltime, etc. If specific nodes are desired to run on, then resource attributes must be defined.

If you would like to get a full dump of your Torque server and queue configuration, you can issue this command:

```
# qmgr -c "print server"
```

The `qmgr` interface can be used to define additional queues and their parameters. You can also change the parameters on the default OSCAR queue, "`workq`". For example, to show the configuration of the `workq`, execute the following:

```
# qmgr -c "list queue workq"
```

To change any of the values listed, use the following:

```
# qmgr -c "set queue workq PARAMETER = VALUE"
```

where `PARAMETER` is a parameter from the "`list queue`" command, and `VALUE` is a valid value for that parameter. You can use the "`print server`" and/or "`list queue`" commands to verify your changes.

Be aware that if you call the `post_install` command with the `--default` option, you will lose your customizations. Also note that OSCAR's default wallclock limit on `workq` is 10,000 hours. Depending on the application mix that will run on your cluster, you may wish to adjust this value.

Some useful links:

- Torque: `http://www.clusterresources.com/products/torque/`
- Maui Scheduler: `http://www.clusterresources.com/products/maui/`
- OpenPBS: `http://www.openpbs.org/`
- PBSPro: `http://www.pbspro.com/`

A Network Booting Client Nodes

There are two methods available for network booting your client nodes. The first is to use the Preboot eXecution Environment (PXE) network boot option in the client's BIOS, if available. If the option is not available, you will need to create a network boot floppy disk using the SystemImager boot package. Each method is described below.

1. **Network booting using PXE.** To use this method, the BIOS and network adapter on each of the client nodes will need to support PXE version 2.0 or later. The PXE specification is available at `http://developer.intel.com/ial/wfm/tools/pxepdk20/`. Earlier versions may work, but experience has shown that versions earlier than 2.0 are unreliable. As BIOS designs vary, there is not a standard procedure for network booting client nodes using PXE. More often than not, the option is presented in one of two ways.

   (a) The first is that the option can be specified in the BIOS boot order list. If presented in the boot order list, you will need to set the client to have network boot as the first boot device. In addition, when you have completed the client installation, remember to reset the BIOS and remove network boot from the boot list so that the client will boot from its local hard drive and will not attempt to do the installation again.

   (b) The second is that the user must watch the output of the client node while booting and press a specified key such as "N" at the appropriate time. In this case, you will need to do so for each client as it boots.

2. **Network booting using a SystemImager boot floppy.** The SystemImager boot package is provided with OSCAR just in case your machines do not have a BIOS network boot option. You can create a boot floppy through the OSCAR GUI installation wizard on the <**`Setup Networking`**> panel or by using the `mkautoinstalldiskette` command.

   Once you have created the SystemImager boot floppy, set your client's BIOS to boot from the floppy drive. Insert the floppy and boot the machine to start the network boot. Check the output for errors to make sure your network boot floppy is working properly. Remember to remove the floppy when you reboot the clients after installation.

B What Happens During Client Installation

Once the client is network booted, it either boots off the autoinstall Diskette that you created or uses PXE to network boot, and loads the install kernel. It then broadcasts a BOOTP/DHCP request to obtain the IP address associated with its MAC address. The DHCP server provides the IP information and the client looks for its auto-install script in `/var/lib/systemimager/scripts/`. The script is named <`nodename`>`.sh` and is a symbolic link to the script for the desired image. The auto-install script is the installation workhorse, and does the following:

1. partitions the disk as specified in the image in `<imagedir>/etc/systemimager/partitionschemes`.

2. mounts the newly created partitions on `/a`.

3. chroots to `/a` and uses `rsync` to bring over all the files in the image.

4. invokes `systemconfigurator` to customize the image to the client's particular hardware and configuration.

5. unmounts `/a`.

Once clone completes, the client will either reboot, halt, or beep as specified when defining the image.

C Installing without a DHCP server

This section provides information on installing OSCAR without the use of a DHCP server on the server node. For most people, this information is not applicable. In addition, this section only presents information where the installation differs from the standard OSCAR install - information presented elsewhere in this document is applicable unless stated otherwise.

Please note that installing an OSCAR cluster with these instructions is not generally supported by the OSCAR developers. While at least one group in the OSCAR team regularly installs and maintains a cluster configured as described here, that doesn't mean it will work in every release. In other words, continue at your own risk.

C.1 Boot Floppies

PXE boot and installation only works correctly if the server running the tftpboot server is also running the DHCP server. As this is obviously not the case if the OSCAR server node can not run a DHCP server, client nodes must be installed using a boot floppy. Further, the boot floppies created with the "Make Boot Floppy" option in the OSCAR wizard assumes there is a DHCP server running on the server node, so we must use a special boot floppy.

Unfortunately, we have to make a boot floppy for every node (since the boot floppy will have IP information encoded in it)[8]. First, a configuration file that provides SIS with the needed boot information must be created. The sample below is used to create a machine `thumb1.osl.iu.edu`. The fields are self-explanatory. The `IMAGESERVER` should be set to the IP address of the OSCAR server node (where you are running the OSCAR wizard). The other networking information must match the information given when building client image during the installation, otherwise your system might not be installable.

```
HOSTNAME=thumb1
DOMAINNAME=osl.iu.edu
DEVICE=eth0
IPADDR=129.79.247.11
NETMASK=255.255.252.0
NETWORK=129.79.244.0
BROADCAST=129.79.247.255
GATEWAY=129.79.247.254
IMAGESERVER=129.79.247.10
```

The boot floppy can be created by running `mkautoinstalldiskette` with the `-config FILE` option specifying the location of the configuration file created above.

[8]This process can obviously be automated, but we do not go into detail on that process here.

C.2 Differences from Standard OSCAR Install

C.2.1 Cluster Definition

When setting up the networking in the OSCAR installation wizard, it is not necessary to collect the MAC addresses of your client nodes. However, to make sure all the proper steps of the OSCAR installation are performed, you should ensure that you open the Wizard pane and click on the "Done" option.

C.2.2 Adding and Deleting client nodes

The Add / Delete client node functionality has not been well tested in this configuration. While there should not be any problems in adding a node to the cluster using boot floppies [9], please be cautious when adding a new node to the cluster.

D Security

D.1 Security layers

Linux cluster security should, ideally, consist of multiple layers. The main security layers are router packet filtering, network stack protections, host based packet filtering, TCP wrappers, service paring, service configuration, and secure communications. Oscar installs a host based packet filtering package called `pfilter`.

D.2 Router packet filtering

This involves adding packet filtering rules to your border network router. Normally, this is not done because of the difficulty in modifying router tables. Packet filtering involves looking at each network packet, and deciding whether each packet should be allowed, dropped, or rejected, based on tables of rules.

D.3 Network stack protections

Linux kernels have security features built in that can help prevent outsiders from pretending that they are part of your internal network. These features are enabled through `/proc` filesystem entries. A good firewall package will turn these on appropriately (`pfilter` does this when enabled).

D.4 Host based packet filtering

Like router packet filtering, host based network packet filtering involves examining each packet and deciding what do do with it. But with host based filtering, each machine individually filters the network packets going to, from, or through it. Linux kernels from 2.4 on include support for connection tracking and "statefull" packet filtering, which keeps track of ongoing network connections, allowing better filtering decisions to be made based on whether packets are part of an already allowed connection.

The problem with packet filtering is that it requires generating filtering "rulesets" that the `iptables` or `ipchains` programs interpret and store in the running kernel. Creating these rulesets is similar to writing software in assembly language. There are now higher level "languages" and compilers that can be used to generate the rulesets and provide firewalls. OSCAR installs a ruleset compiler/firewall package called `pfilter`. For more information, see the `http://pfilter.sourceforge.net/`.

[9] Just remember to create another boot floppy

D.5 TCP Wrappers

TCP wrappers are an access control system that allows control over which network addresses or address ranges can access particular network services on a computer host. This is controlled by the `/etc/hosts.allow` and `/etc/hosts.deny` files. This allows certain services to be only accessible from your local domain, for instance. A common use of this would be to limit exported NFS filesystems to only be accessible from your local domain, while allowing security logins through `ssh` to come in from anywhere. This would be done with a `/etc/hosts.deny` file that looks like this:

```
ALL:   ALL
```

and a `/etc/hosts.allow` file that looks like this:

```
# allow NFS service to domain.net only
portmap:      .domain.net
rpc.mountd:   .domain.net
# allow ssh logins from anywhere
sshd:         ALL
```

D.6 Service paring

This is probably the most used of all the security layers, since turning off unneeded network services gets rid of opportunities for network breakins. To hunt down and turn off unwanted services, the `lsof`, `chkconfig`, and `service` system commands can be used. To display which network services are currently listening on a system, do this:

```
# lsof -i | grep LISTEN | awk '{print $1,$(NF-2),$(NF-1)}' | sort | uniq
```

To list the services that will be started by default at the current run level do this:

```
# chkconfig --list | grep `grep :initdefault: /etc/inittab | \
  awk -F: '{print $2}'`:on | awk '{print $1}' | sort | column
```

To find services started by `xinetd` do this:

```
# chkconfig --list | awk 'NF==2&&$2==''off''{print}' | \
  awk -F: '{print $1}' | sort | column
```

The `nmap` port scanning command is also useful to get a hackers-eye view of your systems. The `chkconfig` and `service` commands can be used to turn on and off system services.

D.7 Service configuration

Some network services have their own configuration files. These should be edited to tighten down outside access. For example, the NFS filesystem uses the `/etc/exports` to determine which network addresses can access individual file systems, and which have read-write or read-only access. Indeed, OSCAR sets `/etc/exports` to only NFS export the `/home` directory to the OSCAR cluster nodes.

D.8 Secure communications

The OSCAR team *strongly* recommends using `ssh` for all network logins, and `scp` to copy files either between OSCAR nodes, or to remote hosts. The use of `telnet`, `rsh`, `ftp`, and other legacy network applications are strongly discouraged because of inherent security risks (passwords are transmitted "in the clear").

`ssh` and `scp` are fully functional replacements for these legacy commands – OSCAR takes care of setting up all the authentication and key management issues for all users.

E Screen-by-Screen Walkthrough

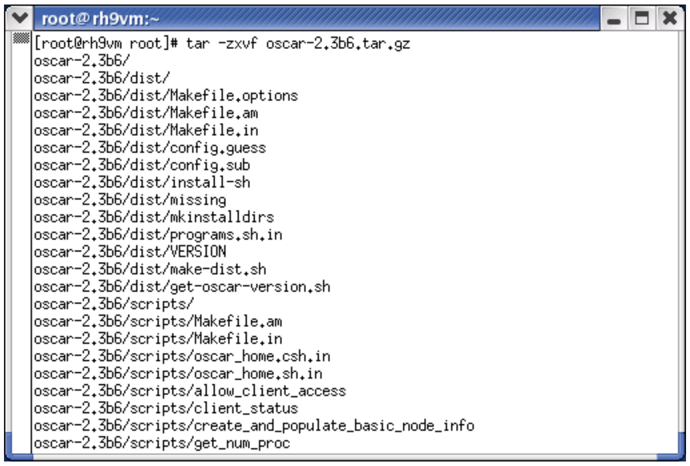
The following is a screen-by-screen walkthrough of a simple installation. It is intended as supplementary material to aid in providing a better feel for the general progression of the installation. For a detailed discussion of the steps, please refer to the Detailed Cluster Installation Procedure.

Note the example screen shots were based in a Red Hat 9 using a pre-release version of OSCAR 4.1 in the GNOME graphical environment. Since this section is intended as a supplementary source of information, it is judged to be "close enough" to the real 4.1 release.

Also note that these images have been scaled down to fit within the document. As such, although they are readable, the images may not render nicely on a screen. The images tend be much more readable on an actual printout.

E.1 Running `install_cluster`

These Figures 10 – 12 walk through the steps needed to prepare and run the `install_cluster` script with the network interface name, which will begin the actual cluster installation. See details in Section 5.2, page 19.



```
[root@rh9vm root]# tar -zxvf oscar-2.3b6.tar.gz
oscar-2.3b6/
oscar-2.3b6/dist/
oscar-2.3b6/dist/Makefile.options
oscar-2.3b6/dist/Makefile.am
oscar-2.3b6/dist/Makefile.in
oscar-2.3b6/dist/config.guess
oscar-2.3b6/dist/config.sub
oscar-2.3b6/dist/install-sh
oscar-2.3b6/dist/missing
oscar-2.3b6/dist/mkinstalldirs
oscar-2.3b6/dist/programs.sh.in
oscar-2.3b6/dist/VERSION
oscar-2.3b6/dist/make-dist.sh
oscar-2.3b6/dist/get-oscar-version.sh
oscar-2.3b6/scripts/
oscar-2.3b6/scripts/Makefile.am
oscar-2.3b6/scripts/Makefile.in
oscar-2.3b6/scripts/oscar_home.csh.in
oscar-2.3b6/scripts/oscar_home.sh.in
oscar-2.3b6/scripts/allow_client_access
oscar-2.3b6/scripts/client_status
oscar-2.3b6/scripts/create_and_populate_basic_node_info
oscar-2.3b6/scripts/get_num_proc
```

Figure 10: Unpacking OSCAR.

E.2 Download Additional OSCAR Packages...

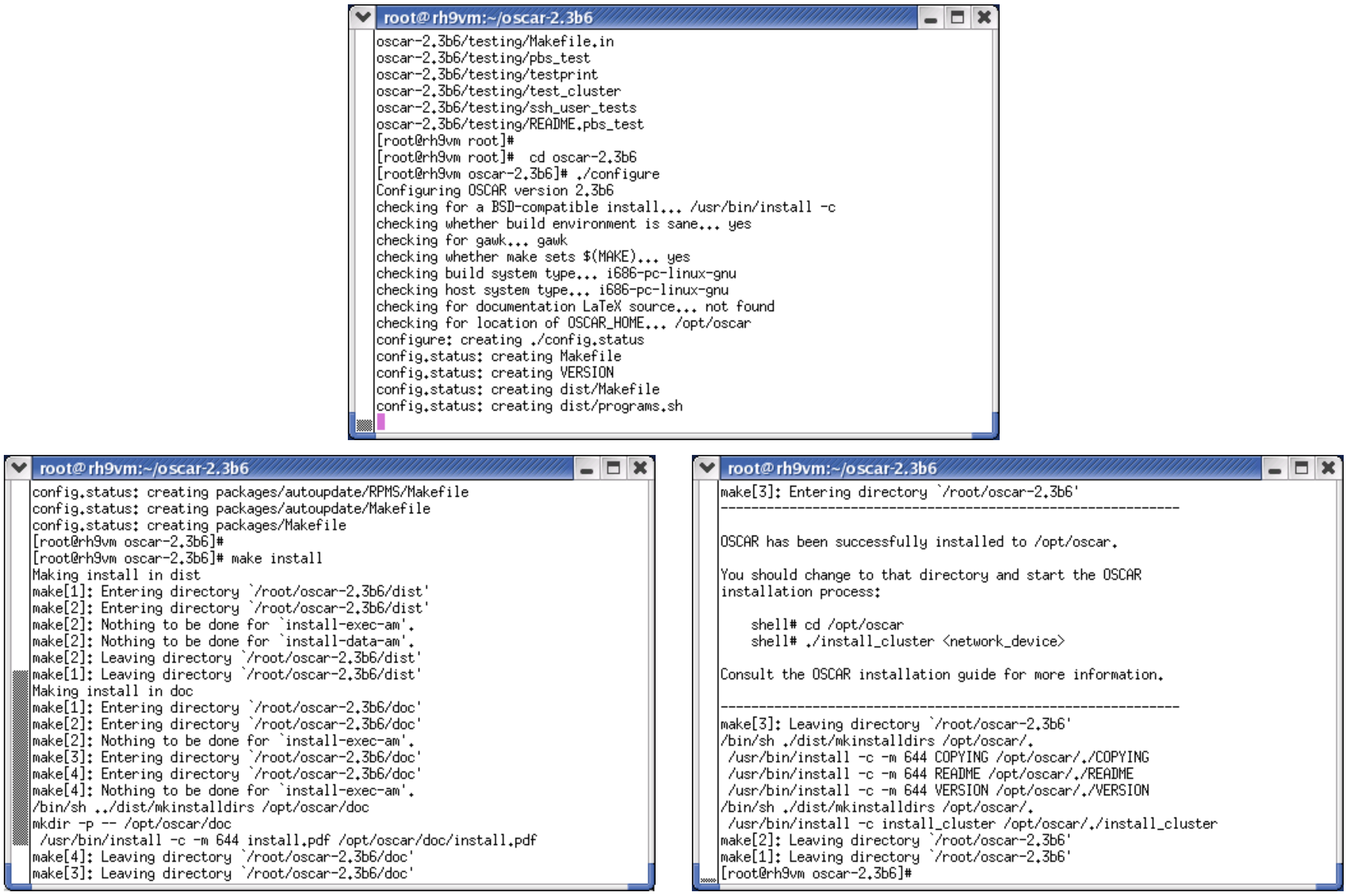
An optional step. See details in Section 5.3, page 20.

E.3 Select OSCAR Packages to Install

An optional step. See details in Section 5.4, page 21.

E.4 Configure Selected OSCAR Packages

An optional step. See details in Section 5.5, page 22.



```
oscar-2.3b6/testing/Makefile.in
oscar-2.3b6/testing/pbs_test
oscar-2.3b6/testing/testprint
oscar-2.3b6/testing/test_cluster
oscar-2.3b6/testing/ssh_user_tests
oscar-2.3b6/testing/README.pbs_test
[root@rh9vm root]#
[root@rh9vm root]#  cd oscar-2.3b6
[root@rh9vm oscar-2.3b6]# ./configure
Configuring OSCAR version 2.3b6
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking build system type... i686-pc-linux-gnu
checking host system type... i686-pc-linux-gnu
checking for documentation LaTeX source... not found
checking for location of OSCAR_HOME... /opt/oscar
configure: creating ./config.status
config.status: creating Makefile
config.status: creating VERSION
config.status: creating dist/Makefile
config.status: creating dist/programs.sh
```

```
config.status: creating packages/autoupdate/RPMS/Makefile
config.status: creating packages/autoupdate/Makefile
config.status: creating packages/Makefile
[root@rh9vm oscar-2.3b6]#
[root@rh9vm oscar-2.3b6]# make install
Making install in dist
make[1]: Entering directory `/root/oscar-2.3b6/dist'
make[2]: Entering directory `/root/oscar-2.3b6/dist'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
make[2]: Leaving directory `/root/oscar-2.3b6/dist'
make[1]: Leaving directory `/root/oscar-2.3b6/dist'
Making install in doc
make[1]: Entering directory `/root/oscar-2.3b6/doc'
make[2]: Entering directory `/root/oscar-2.3b6/doc'
make[2]: Nothing to be done for `install-exec-am'.
make[3]: Entering directory `/root/oscar-2.3b6/doc'
make[4]: Entering directory `/root/oscar-2.3b6/doc'
make[4]: Nothing to be done for `install-exec-am'.
/bin/sh ../dist/mkinstalldirs /opt/oscar/doc
mkdir -p -- /opt/oscar/doc
 /usr/bin/install -c -m 644 install.pdf /opt/oscar/doc/install.pdf
make[4]: Leaving directory `/root/oscar-2.3b6/doc'
make[3]: Leaving directory `/root/oscar-2.3b6/doc'
```

```
make[3]: Entering directory `/root/oscar-2.3b6'
------------------------------------------------------------

OSCAR has been successfully installed to /opt/oscar.

You should change to that directory and start the OSCAR
installation process:

    shell# cd /opt/oscar
    shell# ./install_cluster <network_device>

Consult the OSCAR installation guide for more information.

------------------------------------------------------------
make[3]: Leaving directory `/root/oscar-2.3b6'
/bin/sh ./dist/mkinstalldirs /opt/oscar/.
 /usr/bin/install -c -m 644 COPYING /opt/oscar/./COPYING
 /usr/bin/install -c -m 644 README /opt/oscar/./README
 /usr/bin/install -c -m 644 VERSION /opt/oscar/./VERSION
/bin/sh ./dist/mkinstalldirs /opt/oscar/.
 /usr/bin/install -c install_cluster /opt/oscar/./install_cluster
make[2]: Leaving directory `/root/oscar-2.3b6'
make[1]: Leaving directory `/root/oscar-2.3b6'
[root@rh9vm oscar-2.3b6]#
```

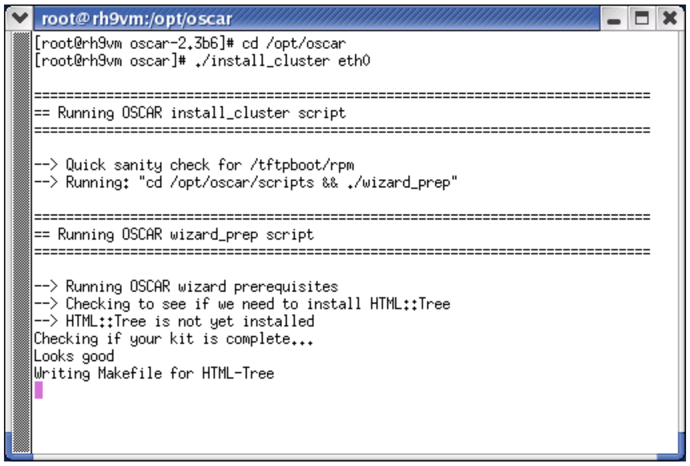
Figure 11: Configure & Install the OSCAR Toolkit.

E.5 Install OSCAR Server Packages

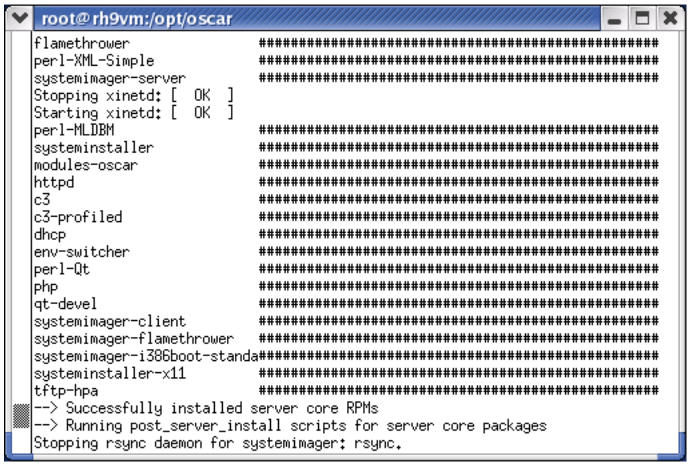
This step is used to setup the server for the OSCAR cluster. See details in Section 5.6, page 23.

E.6 Build OSCAR Client Image

This step builds a disk image for the clients to download and install onto their local disks. See the details in Section 5.7, page 23.

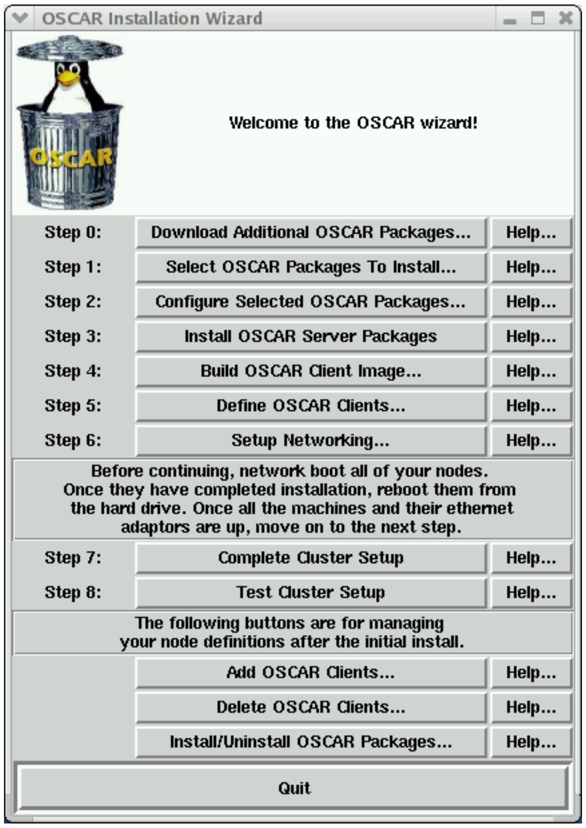


```
root@rh9vm:/opt/oscar
[root@rh9vm oscar-2.3b6]# cd /opt/oscar
[root@rh9vm oscar]# ./install_cluster eth0

==============================================================================
== Running OSCAR install_cluster script
==============================================================================

--> Quick sanity check for /tftpboot/rpm
--> Running: "cd /opt/oscar/scripts && ./wizard_prep"

==============================================================================
== Running OSCAR wizard_prep script
==============================================================================

--> Running OSCAR wizard prerequisites
--> Checking to see if we need to install HTML::Tree
--> HTML::Tree is not yet installed
Checking if your kit is complete...
Looks good
Writing Makefile for HTML-Tree
```



```
root@rh9vm:/opt/oscar
flamethrower           ##################################################
perl-XML-Simple        ##################################################
systemimager-server    ##################################################
Stopping xinetd: [  OK  ]
Starting xinetd: [  OK  ]
perl-MLDBM             ##################################################
systeminstaller        ##################################################
modules-oscar          ##################################################
httpd                  ##################################################
c3                     ##################################################
c3-profiled            ##################################################
dhcp                   ##################################################
env-switcher           ##################################################
perl-Qt                ##################################################
php                    ##################################################
qt-devel               ##################################################
systemimager-client    ##################################################
systemimager-flamethrower ##################################################
systemimager-i386boot-standa##################################################
systeminstaller-x11    ##################################################
tftp-hpa               ##################################################
--> Successfully installed server core RPMs
--> Running post_server_install scripts for server core packages
Stopping rsync daemon for systemimager: rsync.
```

Figure 12: Running the `install_cluster` script.



OSCAR Installation Wizard

Welcome to the OSCAR wizard!

Step 0:	Download Additional OSCAR Packages...	Help...
Step 1:	Select OSCAR Packages To Install...	Help...
Step 2:	Configure Selected OSCAR Packages...	Help...
Step 3:	Install OSCAR Server Packages	Help...
Step 4:	Build OSCAR Client Image...	Help...
Step 5:	Define OSCAR Clients...	Help...
Step 6:	Setup Networking...	Help...

Before continuing, network boot all of your nodes.
Once they have completed installation, reboot them from
the hard drive. Once all the machines and their ethernet
adaptors are up, move on to the next step.

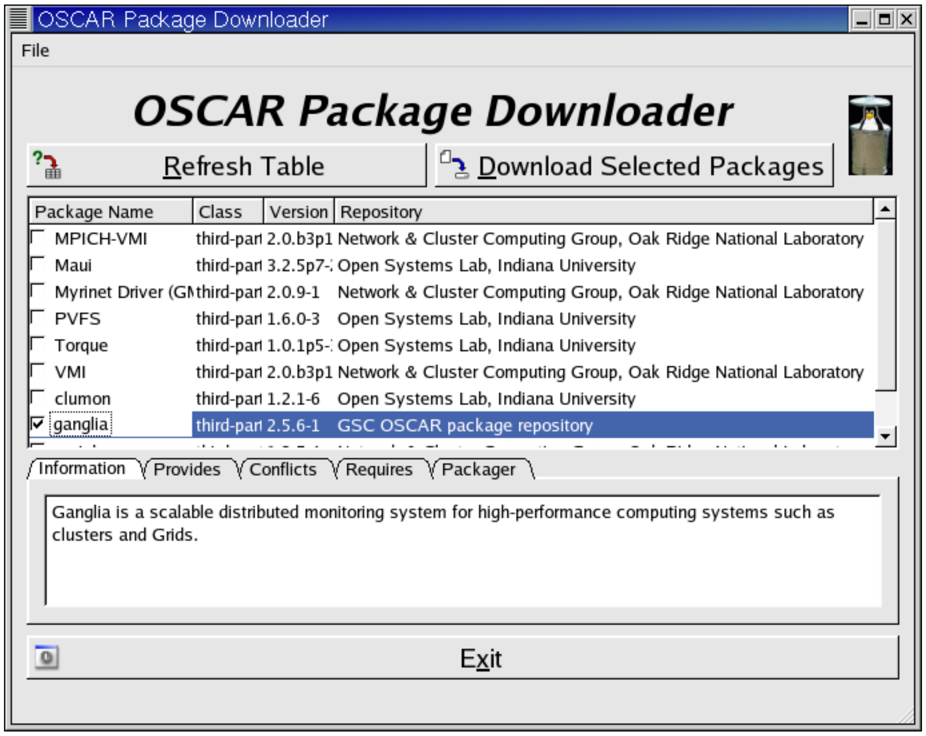
Step 7:	Complete Cluster Setup	Help...
Step 8:	Test Cluster Setup	Help...

The following buttons are for managing
your node definitions after the initial install.

Add OSCAR Clients...	Help...
Delete OSCAR Clients...	Help...
Install/Uninstall OSCAR Packages...	Help...

Quit

Figure 13: The OSCAR Installation Wizard.


OSCAR Package Downloader

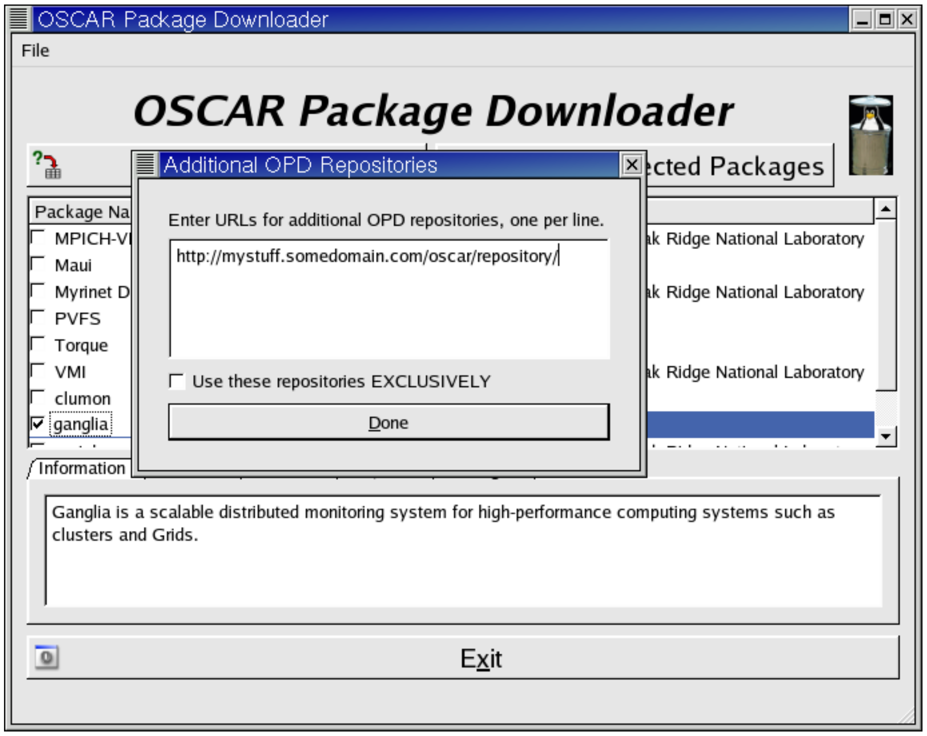
File

OSCAR Package Downloader

	Package Name	Class	Version	Repository
☐	MPICH-VMI	third-part	2.0.b3p1	Network & Cluster Computing Group, Oak Ridge National Laboratory
☐	Maui	third-part	3.2.5p7-:	Open Systems Lab, Indiana University
☐	Myrinet Driver (GM	third-part	2.0.9-1	Network & Cluster Computing Group, Oak Ridge National Laboratory
☐	PVFS	third-part	1.6.0-3	Open Systems Lab, Indiana University
☐	Torque	third-part	1.0.1p5-:	Open Systems Lab, Indiana University
☐	VMI	third-part	2.0.b3p1	Network & Cluster Computing Group, Oak Ridge National Laboratory
☐	clumon	third-part	1.2.1-6	Open Systems Lab, Indiana University
☑	ganglia	third-part	2.5.6-1	GSC OSCAR package repository

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids.

Figure 14: Selecting additional OSCAR packages to download using OPD/OPDer.



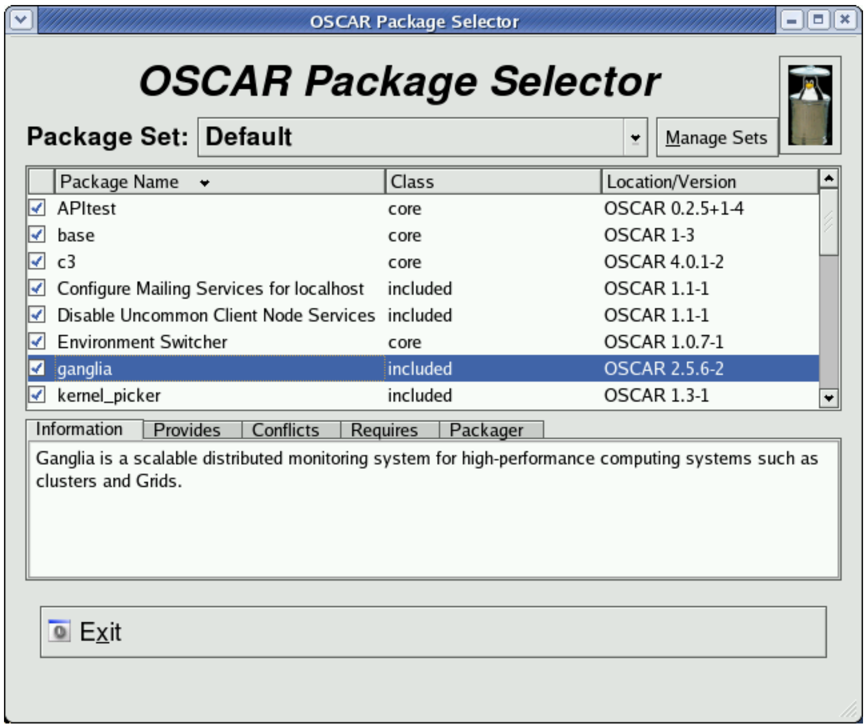
Enter URLs for additional OPD repositories, one per line.

http://mystuff.somedomain.com/oscar/repository/

☐ Use these repositories EXCLUSIVELY

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids.

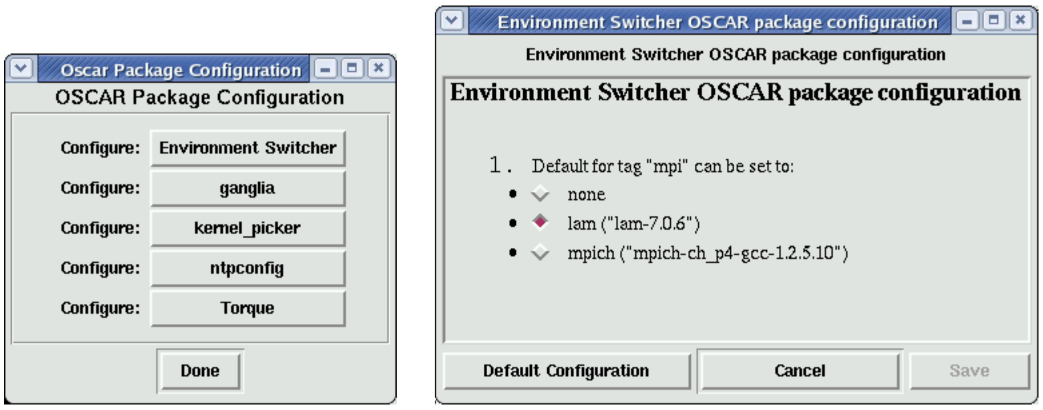
Figure 15: Adding Additional OPD Repositories.



✓	Package Name	Class	Location/Version
✓	APItest	core	OSCAR 0.2.5+1-4
✓	base	core	OSCAR 1-3
✓	c3	core	OSCAR 4.0.1-2
✓	Configure Mailing Services for localhost	included	OSCAR 1.1-1
✓	Disable Uncommon Client Node Services	included	OSCAR 1.1-1
✓	Environment Switcher	core	OSCAR 1.0.7-1
✓	ganglia	included	OSCAR 2.5.6-2
✓	kernel_picker	included	OSCAR 1.3-1

Ganglia is a scalable distributed monitoring system for high-performance computing systems such as clusters and Grids.

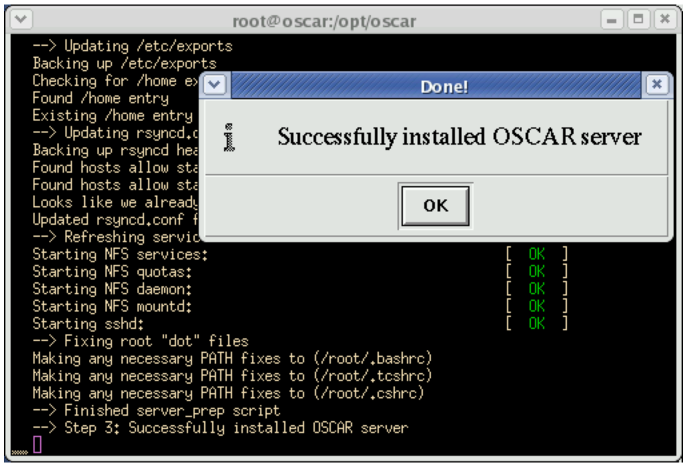
Figure 16: Selecting which OSCAR packages to install.


Oscar Package Configuration
OSCAR Package Configuration
Configure: Environment Switcher
Configure: ganglia
Configure: kernel_picker
Configure: ntpconfig
Configure: Torque
Done
Environment Switcher OSCAR package configuration
Environment Switcher OSCAR package configuration
Environment Switcher OSCAR package configuration

1. Default for tag "mpi" can be set to:
   - none
   - [✓ lam ("lam-7.0.6")]
   - mpich ("mpich-ch_p4-gcc-1.2.5.10")

Default Configuration
Cancel
Save

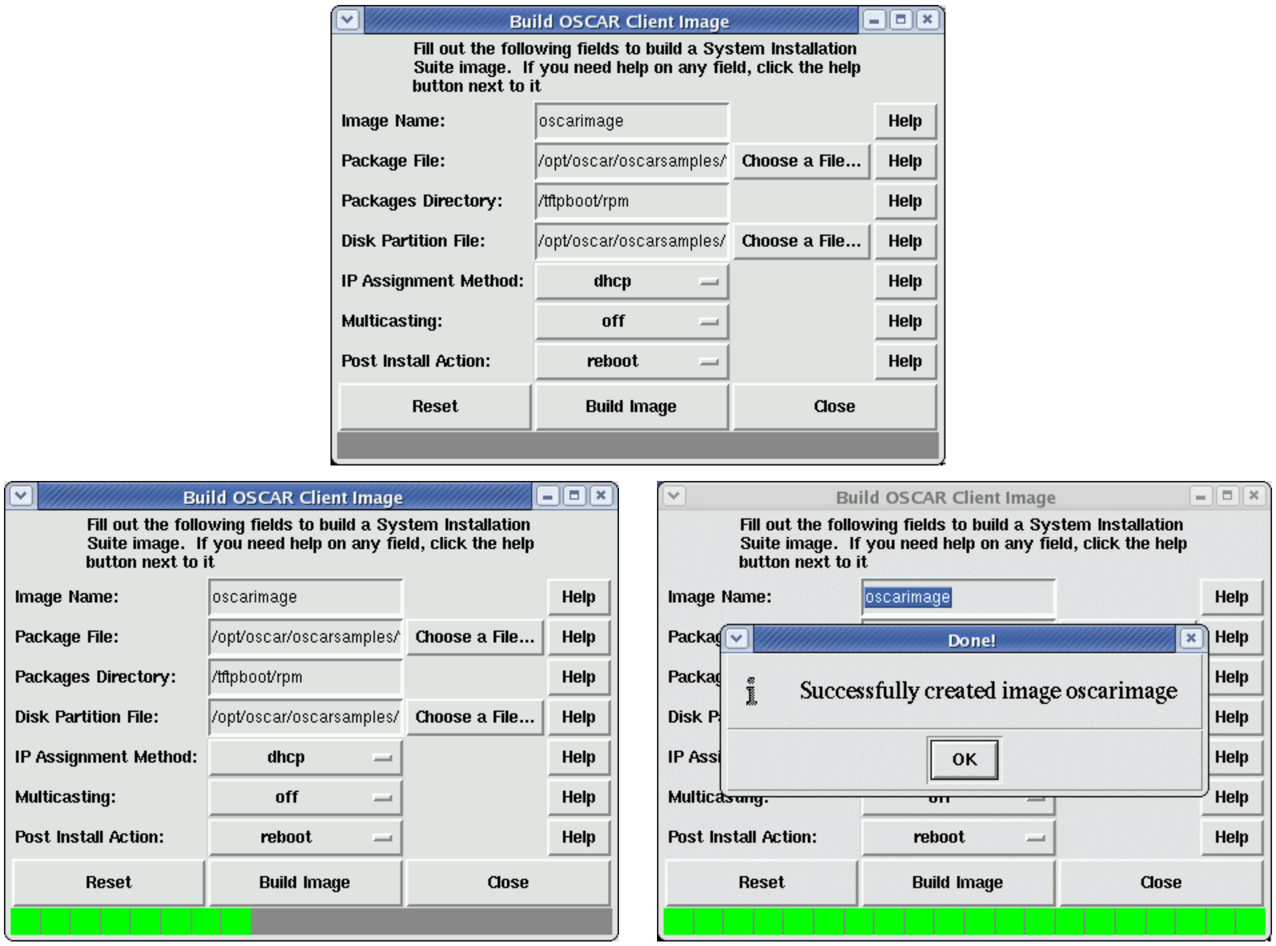
Figure 17: Configuring selected OSCAR packages. For example, the Environment Switcher package has configuration options.



```
--> Updating /etc/exports
Backing up /etc/exports
Checking for /home ex
Found /home entry
Existing /home entry
--> Updating rsyncd.c
Backing up rsyncd hea
Found hosts allow sta
Found hosts allow sta
Looks like we already
Updated rsyncd.conf f
--> Refreshing servic
Starting NFS services:                                    [  OK  ]
Starting NFS quotas:                                      [  OK  ]
Starting NFS daemon:                                      [  OK  ]
Starting NFS mountd:                                      [  OK  ]
Starting sshd:                                            [  OK  ]
--> Fixing root "dot" files
Making any necessary PATH fixes to (/root/.bashrc)
Making any necessary PATH fixes to (/root/.tcshrc)
Making any necessary PATH fixes to (/root/.cshrc)
--> Finished server_prep script
--> Step 3: Successfully installed OSCAR server
```

Successfully installed OSCAR server

Figure 18: Successfully installed the OSCAR server packages.



Fill out the following fields to build a System Installation Suite image. If you need help on any field, click the help button next to it

Image Name:	oscarimage
Package File:	/opt/oscar/oscarsamples/
Packages Directory:	/tftpboot/rpm
Disk Partition File:	/opt/oscar/oscarsamples/
IP Assignment Method:	dhcp
Multicasting:	off
Post Install Action:	reboot

Successfully created image oscarimage

Figure 19: Building the OSCAR client image.

E.7 Define OSCAR Clients

Step 3 is used to specify how many clients there will be, and what their TCP/IP characteristics will be. See the details in Section 5.8, page 25.

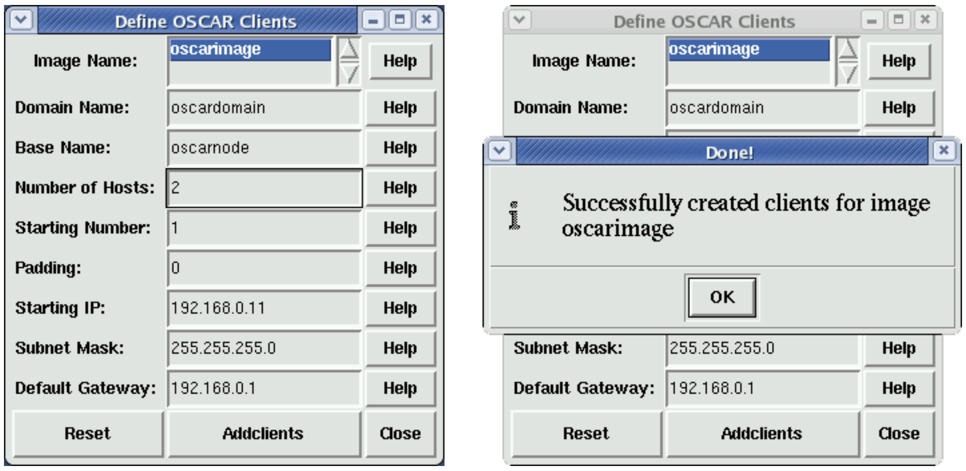

Define OSCAR Clients

Image Name:	oscarimage
Domain Name:	oscardomain
Base Name:	oscarnode
Number of Hosts:	2
Starting Number:	1
Padding:	0
Starting IP:	192.168.0.11
Subnet Mask:	255.255.255.0
Default Gateway:	192.168.0.1

Define OSCAR Clients

Image Name:	oscarimage
Domain Name:	oscardomain
Subnet Mask:	255.255.255.0
Default Gateway:	192.168.0.1

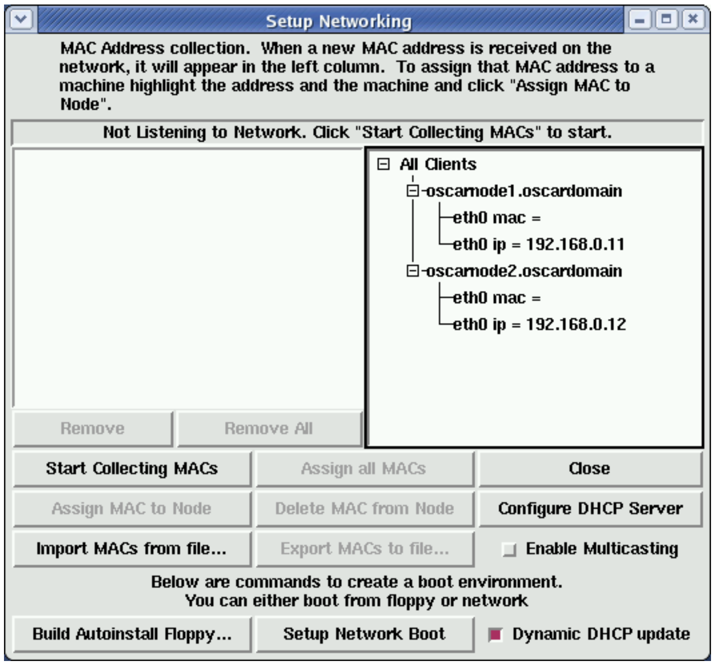
Done!

Successfully created clients for image oscarimage

Figure 20: Defining the OSCAR clients.

E.8 Setup Networking

This step is used to collect the MAC addresses of the clients, and then download the disk images to the clients. See the details in Section 5.9, page 27.



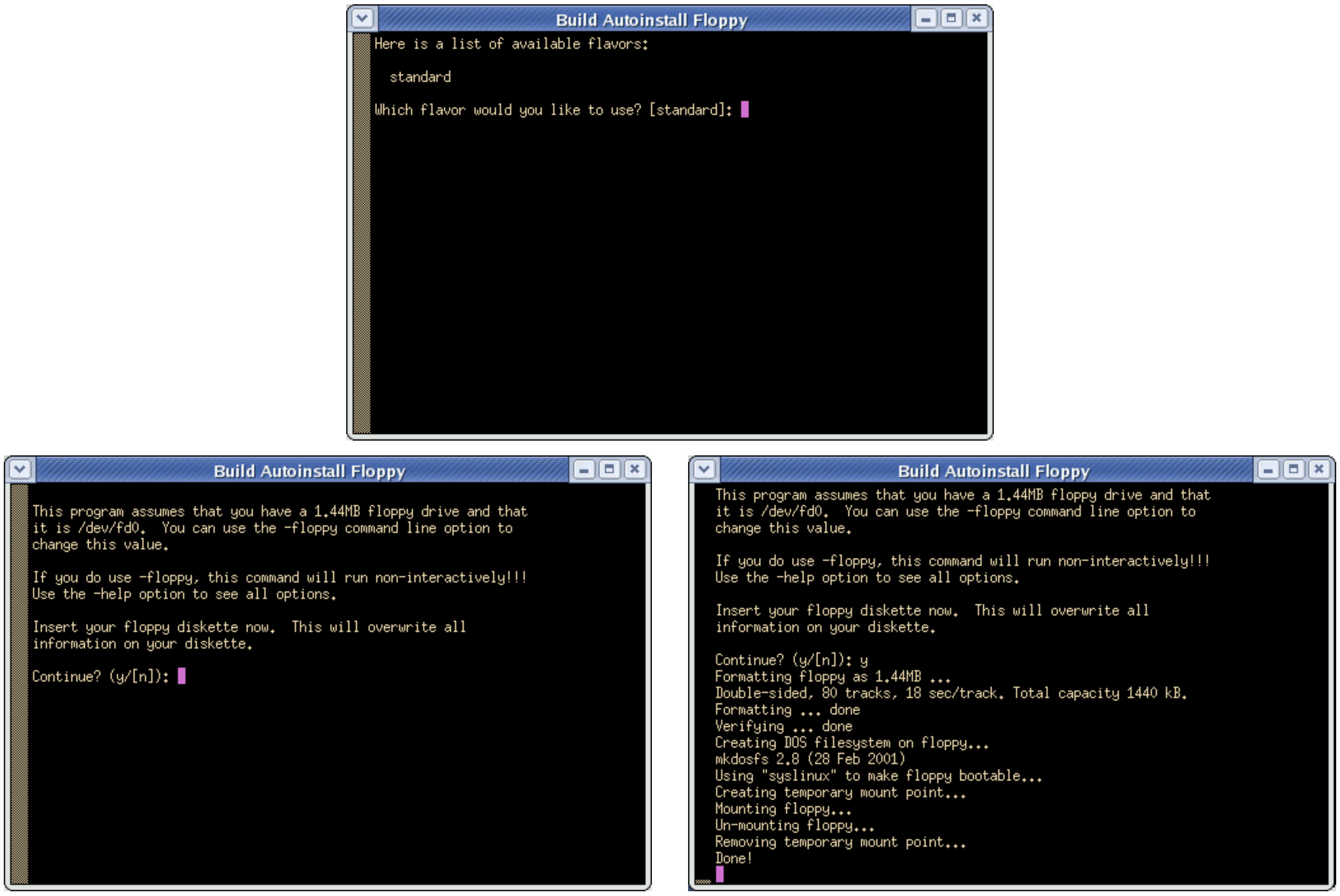
MAC Address collection. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click "Assign MAC to Node".

Not Listening to Network. Click "Start Collecting MACs" to start.

```
All Clients
  oscarnode1.oscardomain
    eth0 mac =
    eth0 ip = 192.168.0.11
  oscarnode2.oscardomain
    eth0 mac =
    eth0 ip = 192.168.0.12
```

Below are commands to create a boot environment.
You can either boot from floppy or network

Figure 21: Setup networking: initial window.

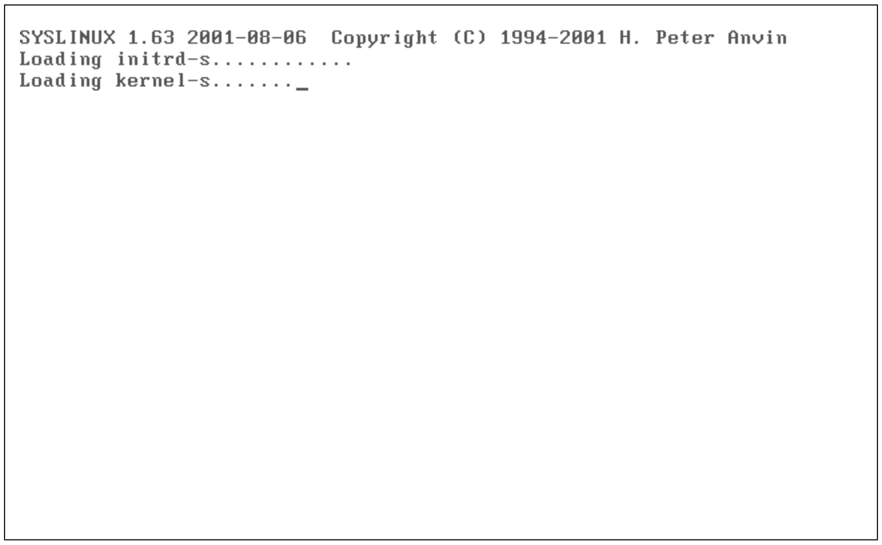


```
Here is a list of available flavors:

  standard

Which flavor would you like to use? [standard]:
```

```
This program assumes that you have a 1.44MB floppy drive and that
it is /dev/fd0.  You can use the -floppy command line option to
change this value.

If you do use -floppy, this command will run non-interactively!!!
Use the -help option to see all options.

Insert your floppy diskette now.  This will overwrite all
information on your diskette.

Continue? (y/[n]):
```

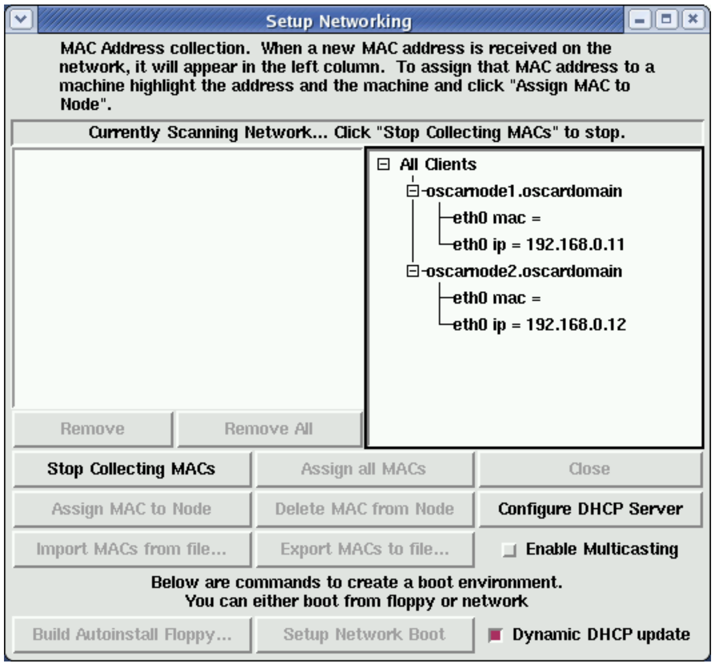
```
This program assumes that you have a 1.44MB floppy drive and that
it is /dev/fd0.  You can use the -floppy command line option to
change this value.

If you do use -floppy, this command will run non-interactively!!!
Use the -help option to see all options.

Insert your floppy diskette now.  This will overwrite all
information on your diskette.

Continue? (y/[n]): y
Formatting floppy as 1.44MB ...
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
Creating DOS filesystem on floppy...
mkdosfs 2.8 (28 Feb 2001)
Using "syslinux" to make floppy bootable...
Creating temporary mount point...
Mounting floppy...
Un-mounting floppy...
Removing temporary mount point...
Done!
```

Figure 22: Setup networking: building an autoinstall floppy.



```
SYSLINUX 1.63 2001-08-06  Copyright (C) 1994-2001 H. Peter Anvin
Loading initrd-s...........
Loading kernel-s......._
```

Figure 23: Setup networking: booting the client.



Setup Networking

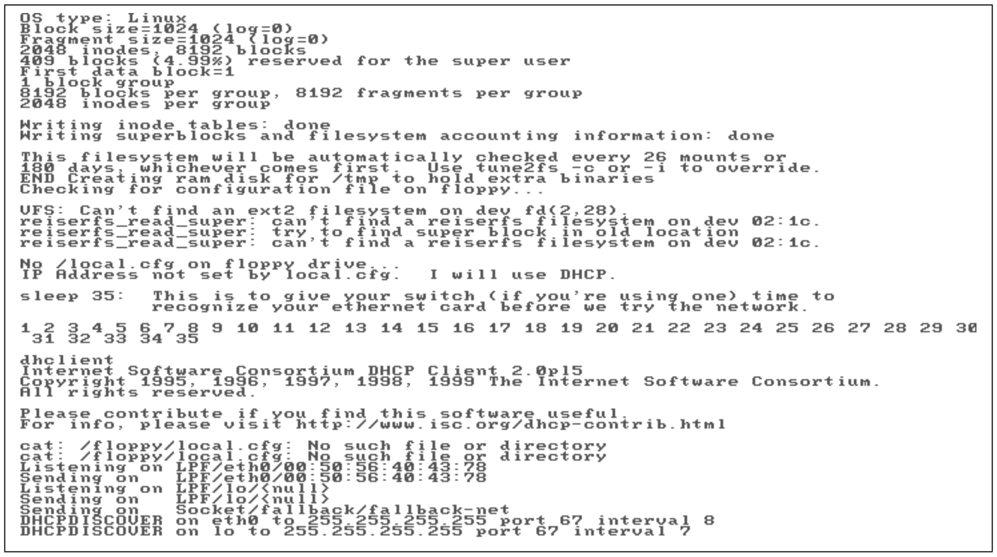
MAC Address collection. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click "Assign MAC to Node".

Currently Scanning Network... Click "Stop Collecting MACs" to stop.

```
⊟ All Clients
   ⊟-oscarnode1.oscardomain
      ├eth0 mac =
      └eth0 ip = 192.168.0.11
   ⊟-oscarnode2.oscardomain
      ├eth0 mac =
      └eth0 ip = 192.168.0.12
```

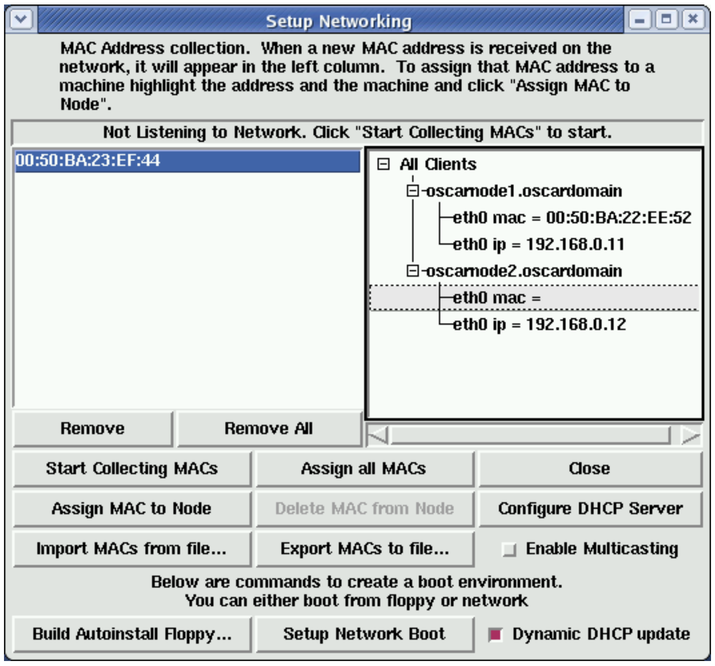
Below are commands to create a boot environment.
You can either boot from floppy or network

Figure 24: Setup networking: scan for booting client (DHCP request).



```
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
2048 inodes, 8192 blocks
409 blocks (4.99%) reserved for the super user
First data block=1
1 block group
8192 blocks per group, 8192 fragments per group
2048 inodes per group

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
END Creating ram disk for /tmp to hold extra binaries
Checking for configuration file on floppy...

VFS: Can't find an ext2 filesystem on dev fd(2,28).
reiserfs_read_super: can't find a reiserfs filesystem on dev 02:1c.
reiserfs_read_super: try to find super block in old location
reiserfs_read_super: can't find a reiserfs filesystem on dev 02:1c.

No /local.cfg on floppy drive...
IP Address not set by local.cfg.  I will use DHCP.

sleep 35:  This is to give your switch (if you're using one) time to
           recognize your ethernet card before we try the network.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
 31 32 33 34 35

dhclient
Internet Software Consortium DHCP Client 2.0pl5
Copyright 1995, 1996, 1997, 1998, 1999 The Internet Software Consortium.
All rights reserved.

Please contribute if you find this software useful.
For info, please visit http://www.isc.org/dhcp-contrib.html

cat: /floppy/local.cfg: No such file or directory
cat: /floppy/local.cfg: No such file or directory
Listening on LPF/eth0/00:50:56:40:43:78
Sending on   LPF/eth0/00:50:56:40:43:78
Listening on LPF/lo/<null>
Sending on   LPF/lo/<null>
Sending on   Socket/fallback/fallback-net
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
DHCPDISCOVER on lo to 255.255.255.255 port 67 interval 7
```

Figure 25: Setup networking: client is broadcasting, allows capture of MAC address.

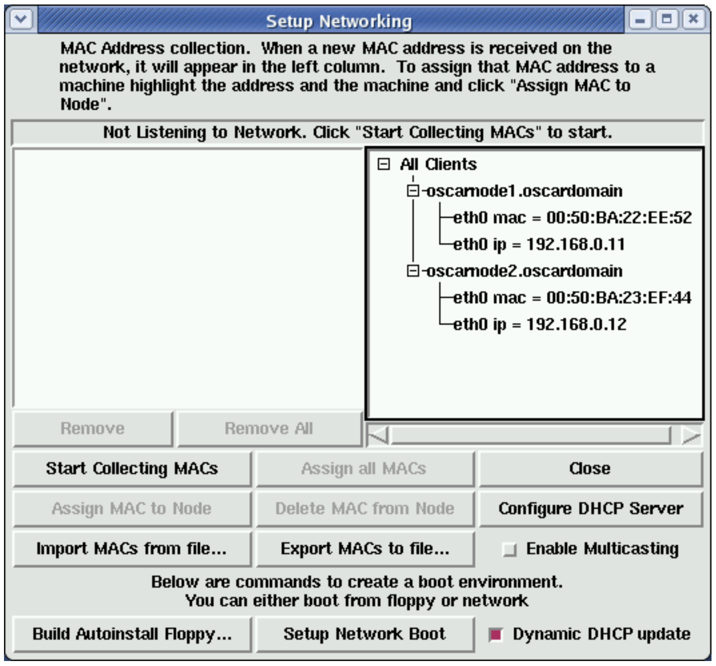


MAC Address collection. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click "Assign MAC to Node".

Not Listening to Network. Click "Start Collecting MACs" to start.

```
00:50:BA:23:EF:44

⊟ All Clients
  ⊟-oscarnode1.oscardomain
      ├eth0 mac = 00:50:BA:22:EE:52
      └eth0 ip = 192.168.0.11
  ⊟-oscarnode2.oscardomain
      ├eth0 mac =
      └eth0 ip = 192.168.0.12
```

Below are commands to create a boot environment.
You can either boot from floppy or network



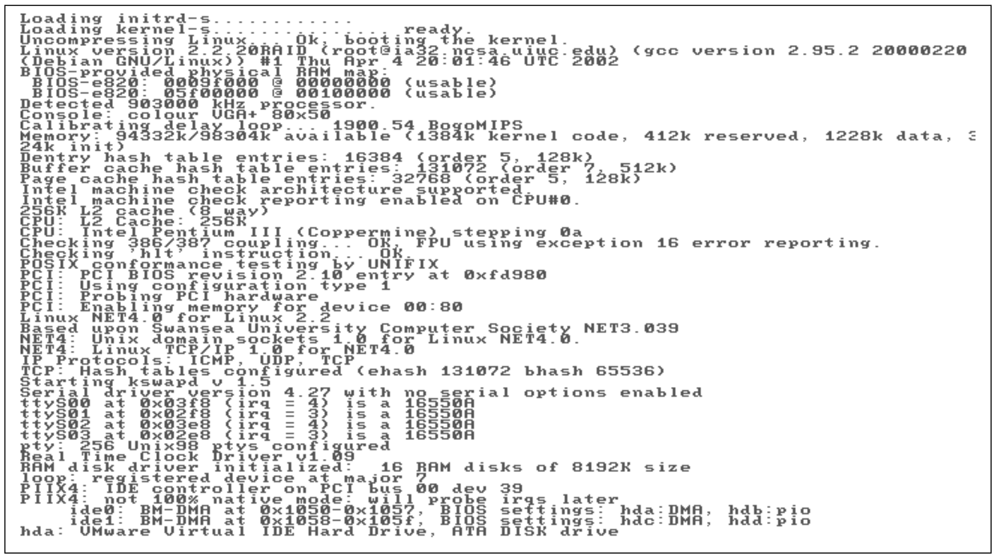
MAC Address collection. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click "Assign MAC to Node".

Not Listening to Network. Click "Start Collecting MACs" to start.

```
⊟ All Clients
  ⊟-oscarnode1.oscardomain
      ├eth0 mac = 00:50:BA:22:EE:52
      └eth0 ip = 192.168.0.11
  ⊟-oscarnode2.oscardomain
      ├eth0 mac = 00:50:BA:23:EF:44
      └eth0 ip = 192.168.0.12
```

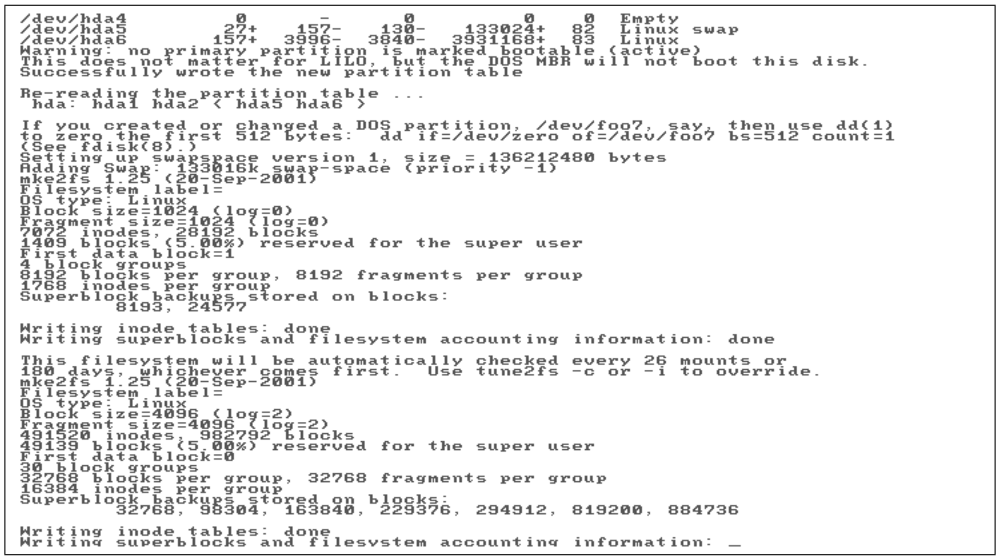
Below are commands to create a boot environment.
You can either boot from floppy or network

Figure 26: Setup networking: Scanning network, found first MAC address, then later assigned all MAC addresses.



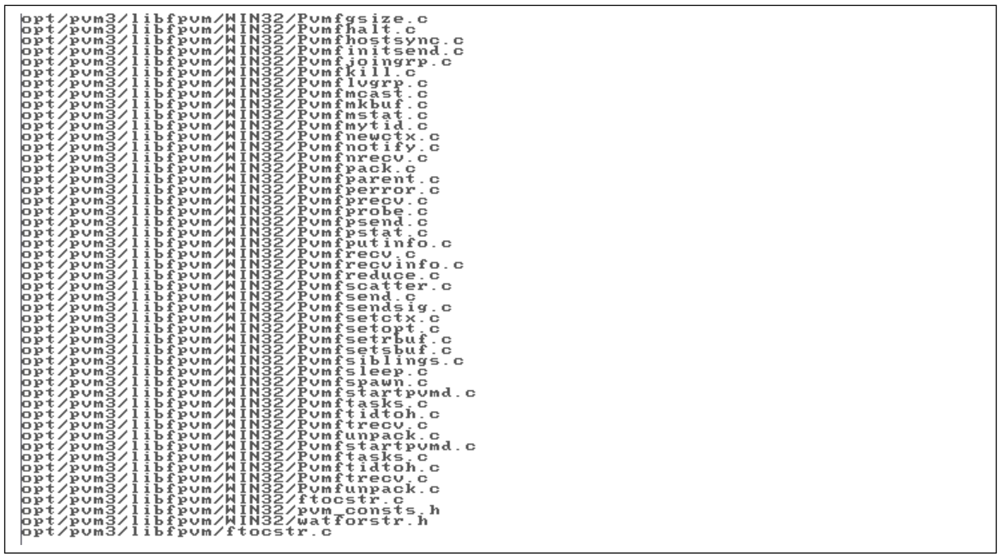
```
Loading initrd-s............
Loading kernel-s.............. ready.
Uncompressing Linux... Ok, booting the kernel.
Linux version 2.2.20RAID (root@ia32.ncsa.uiuc.edu) (gcc version 2.95.2 20000220
(Debian GNU/Linux)) #1 Thu Apr 4 20:01:46 UTC 2002
BIOS-provided physical RAM map:
 BIOS-e820: 0009f000 @ 00000000 (usable)
 BIOS-e820: 05f00000 @ 00100000 (usable)
Detected 903000 kHz processor.
Console: colour VGA+ 80x50
Calibrating delay loop... 1900.54 BogoMIPS
Memory: 94332k/98304k available (1384k kernel code, 412k reserved, 1228k data, 3
24k init)
Dentry hash table entries: 16384 (order 5, 128k)
Buffer cache hash table entries: 131072 (order 7, 512k)
Page cache hash table entries: 32768 (order 5, 128k)
Intel machine check architecture supported.
Intel machine check reporting enabled on CPU#0.
256K L2 cache (8 way)
CPU: L2 Cache: 256K
CPU: Intel Pentium III (Coppermine) stepping 0a
Checking 386/387 coupling... OK, FPU using exception 16 error reporting.
Checking 'hlt' instruction... OK.
POSIX conformance testing by UNIFIX
PCI: PCI BIOS revision 2.10 entry at 0xfd980
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Enabling memory for device 00:80
Linux NET4.0 for Linux 2.2
Based upon Swansea University Computer Society NET3.039
NET4: Unix domain sockets 1.0 for Linux NET4.0.
NET4: Linux TCP/IP 1.0 for NET4.0
IP Protocols: ICMP, UDP, TCP
TCP: Hash tables configured (ehash 131072 bhash 65536)
Starting kswapd v 1.5
Serial driver version 4.27 with no serial options enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
ttyS01 at 0x02f8 (irq = 3) is a 16550A
ttyS02 at 0x03e8 (irq = 4) is a 16550A
ttyS03 at 0x02e8 (irq = 3) is a 16550A
pty: 256 Unix98 ptys configured
Real Time Clock Driver v1.09
RAM disk driver initialized:  16 RAM disks of 8192K size
loop: registered device at major 7
PIIX4: IDE controller on PCI bus 00 dev 39
PIIX4: not 100% native mode: will probe irqs later
    ide0: BM-DMA at 0x1050-0x1057, BIOS settings: hda:DMA, hdb:pio
    ide1: BM-DMA at 0x1058-0x105f, BIOS settings: hdc:DMA, hdd:pio
hda: VMware Virtual IDE Hard Drive, ATA DISK drive
```

Figure 27: Booting the client a second time to download the image.



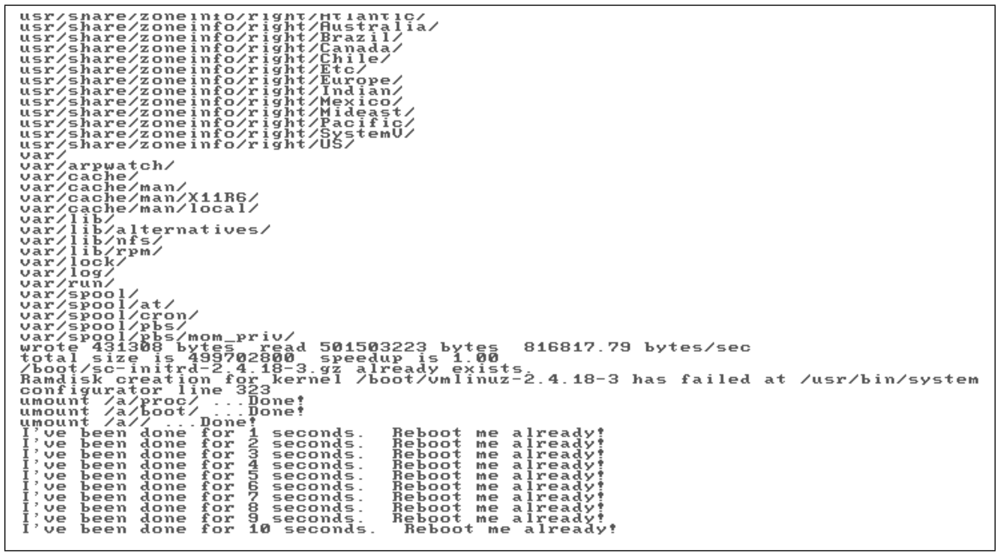
```
/dev/hda4          0      -       0         0    0  Empty
/dev/hda5         27+   157-    130-   133024+  82  Linux swap
/dev/hda6        157+  3996-   3840-  3931168+  83  Linux
Warning: no primary partition is marked bootable (active)
This does not matter for LILO, but the DOS MBR will not boot this disk.
Successfully wrote the new partition table

Re-reading the partition table ...
 hda: hda1 hda2 < hda5 hda6 >

If you created or changed a DOS partition, /dev/foo7, say, then use dd(1)
to zero the first 512 bytes:  dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(See fdisk(8).)
Setting up swapspace version 1, size = 136212480 bytes
Adding Swap: 133016k swap-space (priority -1)
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
7072 inodes, 28192 blocks
1409 blocks (5.00%) reserved for the super user
First data block=1
4 block groups
8192 blocks per group, 8192 fragments per group
1768 inodes per group
Superblock backups stored on blocks:
        8193, 24577

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
mke2fs 1.25 (20-Sep-2001)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
491520 inodes, 982792 blocks
49139 blocks (5.00%) reserved for the super user
First data block=0
30 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Writing superblocks and filesystem accounting information: _
```

Figure 28: Client partitioning disk, setting up the disk tables, and starting to download the image.



```
opt/pvm3/libfpvm/WIN32/Pvmfgsize.c
opt/pvm3/libfpvm/WIN32/Pvmfhalt.c
opt/pvm3/libfpvm/WIN32/Pvmfhostsync.c
opt/pvm3/libfpvm/WIN32/Pvmfinitsend.c
opt/pvm3/libfpvm/WIN32/Pvmfjoingrp.c
opt/pvm3/libfpvm/WIN32/Pvmfkill.c
opt/pvm3/libfpvm/WIN32/Pvmflvgrp.c
opt/pvm3/libfpvm/WIN32/Pvmfmcast.c
opt/pvm3/libfpvm/WIN32/Pvmfmkbuf.c
opt/pvm3/libfpvm/WIN32/Pvmfmstat.c
opt/pvm3/libfpvm/WIN32/Pvmfmytid.c
opt/pvm3/libfpvm/WIN32/Pvmfnewctx.c
opt/pvm3/libfpvm/WIN32/Pvmfnotify.c
opt/pvm3/libfpvm/WIN32/Pvmfnrecv.c
opt/pvm3/libfpvm/WIN32/Pvmfpack.c
opt/pvm3/libfpvm/WIN32/Pvmfparent.c
opt/pvm3/libfpvm/WIN32/Pvmfperror.c
opt/pvm3/libfpvm/WIN32/Pvmfprecv.c
opt/pvm3/libfpvm/WIN32/Pvmfprobe.c
opt/pvm3/libfpvm/WIN32/Pvmfpsend.c
opt/pvm3/libfpvm/WIN32/Pvmfpstat.c
opt/pvm3/libfpvm/WIN32/Pvmfputinfo.c
opt/pvm3/libfpvm/WIN32/Pvmfrecv.c
opt/pvm3/libfpvm/WIN32/Pvmfrecvinfo.c
opt/pvm3/libfpvm/WIN32/Pvmfreduce.c
opt/pvm3/libfpvm/WIN32/Pvmfscatter.c
opt/pvm3/libfpvm/WIN32/Pvmfsend.c
opt/pvm3/libfpvm/WIN32/Pvmfsendsig.c
opt/pvm3/libfpvm/WIN32/Pvmfsetctx.c
opt/pvm3/libfpvm/WIN32/Pvmfsetopt.c
opt/pvm3/libfpvm/WIN32/Pvmfsetrbuf.c
opt/pvm3/libfpvm/WIN32/Pvmfsetsbuf.c
opt/pvm3/libfpvm/WIN32/Pvmfsiblings.c
opt/pvm3/libfpvm/WIN32/Pvmfsleep.c
opt/pvm3/libfpvm/WIN32/Pvmfspawn.c
opt/pvm3/libfpvm/WIN32/Pvmfstartpvmd.c
opt/pvm3/libfpvm/WIN32/Pvmftasks.c
opt/pvm3/libfpvm/WIN32/Pvmftidtoh.c
opt/pvm3/libfpvm/WIN32/Pvmftrecv.c
opt/pvm3/libfpvm/WIN32/Pvmfunpack.c
opt/pvm3/libfpvm/WIN32/Pvmfstartpvmd.c
opt/pvm3/libfpvm/WIN32/Pvmftasks.c
opt/pvm3/libfpvm/WIN32/Pvmftidtoh.c
opt/pvm3/libfpvm/WIN32/Pvmftrecv.c
opt/pvm3/libfpvm/WIN32/Pvmfunpack.c
opt/pvm3/libfpvm/WIN32/ftocstr.c
opt/pvm3/libfpvm/WIN32/pvm_consts.h
opt/pvm3/libfpvm/WIN32/watforstr.h
opt/pvm3/libfpvm/ftocstr.c
```

Figure 29: Client downloading and installing the image.

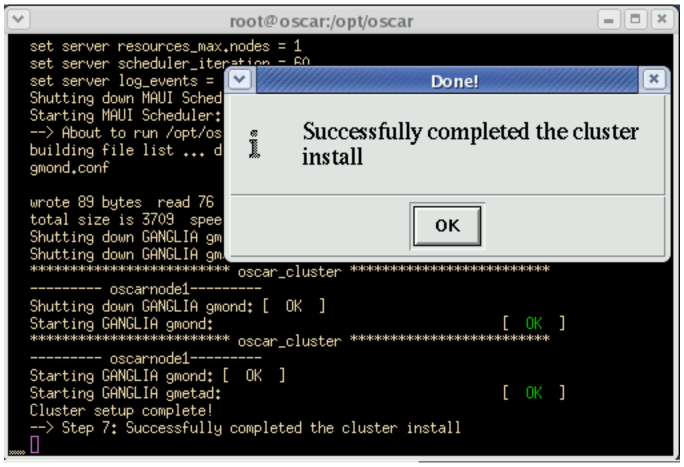


```
usr/share/zoneinfo/right/Atlantic/
usr/share/zoneinfo/right/Australia/
usr/share/zoneinfo/right/Brazil/
usr/share/zoneinfo/right/Canada/
usr/share/zoneinfo/right/Chile/
usr/share/zoneinfo/right/Etc/
usr/share/zoneinfo/right/Europe/
usr/share/zoneinfo/right/Indian/
usr/share/zoneinfo/right/Mexico/
usr/share/zoneinfo/right/Mideast/
usr/share/zoneinfo/right/Pacific/
usr/share/zoneinfo/right/SystemV/
usr/share/zoneinfo/right/US/
var/
var/arpwatch/
var/cache/
var/cache/man/
var/cache/man/X11R6/
var/cache/man/local/
var/lib/
var/lib/alternatives/
var/lib/nfs/
var/lib/rpm/
var/lock/
var/log/
var/run/
var/spool/
var/spool/at/
var/spool/cron/
var/spool/pbs/
var/spool/pbs/mom_priv/
wrote 431308 bytes  read 501503223 bytes  816817.79 bytes/sec
total size is 499702800  speedup is 1.00
/boot/sc-initrd-2.4.18-3.gz already exists.
Ramdisk creation for kernel /boot/vmlinuz-2.4.18-3 has failed at /usr/bin/system
configurator line 323
umount /a/proc/ ...Done!
umount /a/boot/ ...Done!
umount /a// ...Done!
I've been done for 1 seconds.  Reboot me already!
I've been done for 2 seconds.  Reboot me already!
I've been done for 3 seconds.  Reboot me already!
I've been done for 4 seconds.  Reboot me already!
I've been done for 5 seconds.  Reboot me already!
I've been done for 6 seconds.  Reboot me already!
I've been done for 7 seconds.  Reboot me already!
I've been done for 8 seconds.  Reboot me already!
I've been done for 9 seconds.  Reboot me already!
I've been done for 10 seconds.  Reboot me already!
```

Figure 30: A client has finished the install and is asking to be rebooted.

E.9 Step 5: Complete Cluster Setup

This step is used to unify the server and client installations into a single cluster. See the details in Section 5.11, page 29.



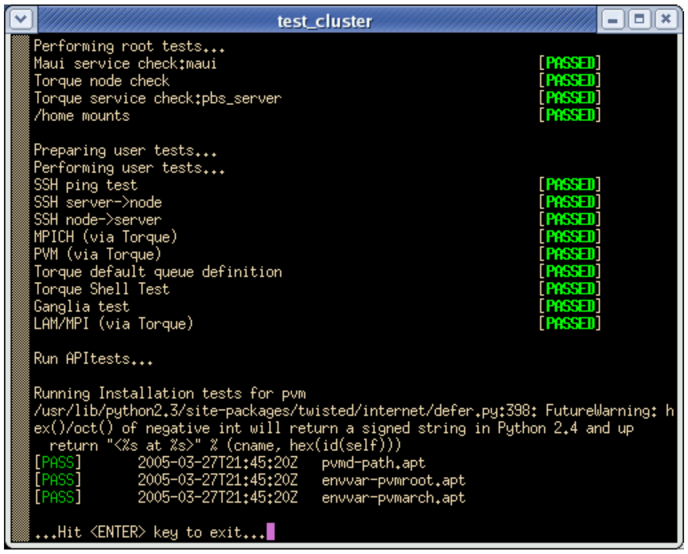
Successfully completed the cluster install

```
set server resources_max.nodes = 1
set server scheduler_iteration = 60
set server log_events =
Shutting down MAUI Sched
Starting MAUI Scheduler:
--> About to run /opt/os
building file list ... d
gmond.conf

wrote 89 bytes  read 76
total size is 3709  spee
Shutting down GANGLIA gm
Shutting down GANGLIA gm
************************* oscar_cluster *************************
--------- oscarnode1---------
Shutting down GANGLIA gmond: [  OK  ]
Starting GANGLIA gmond:                                    [  OK  ]
************************* oscar_cluster *************************
--------- oscarnode1---------
Starting GANGLIA gmond: [  OK  ]
Starting GANGLIA gmetad:                                   [  OK  ]
Cluster setup complete!
--> Step 7: Successfully completed the cluster install
```

Figure 31: Complete Cluster Setup.

E.10 Test Cluster Setup

This step is used to test the cluster setup. It can either be run from within the wizard, or, as shown here, from manually launching a shell script at a `root` command prompt. See the details in Section 5.12, page 29.

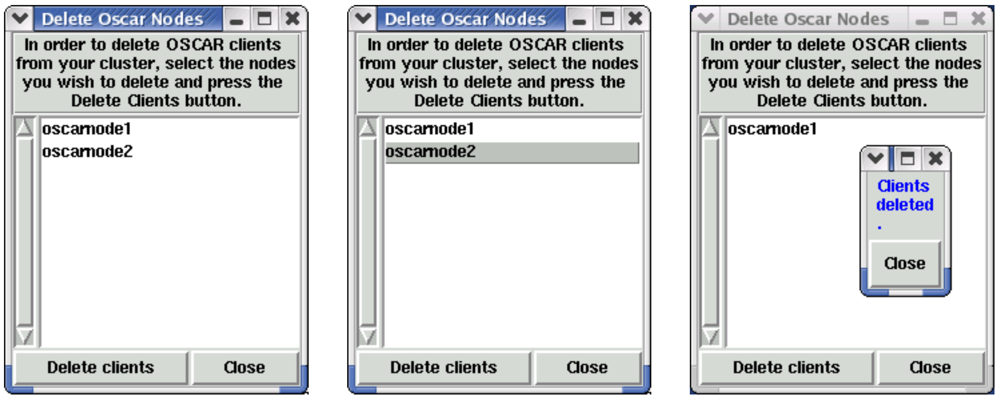


```
Performing root tests...
Maui service check:maui                                        [PASSED]
Torque node check                                              [PASSED]
Torque service check:pbs_server                                [PASSED]
/home mounts                                                   [PASSED]

Preparing user tests...
Performing user tests...
SSH ping test                                                  [PASSED]
SSH server->node                                               [PASSED]
SSH node->server                                               [PASSED]
MPICH (via Torque)                                             [PASSED]
PVM (via Torque)                                               [PASSED]
Torque default queue definition                                [PASSED]
Torque Shell Test                                              [PASSED]
Ganglia test                                                   [PASSED]
LAM/MPI (via Torque)                                           [PASSED]

Run APItests...

Running Installation tests for pvm
/usr/lib/python2.3/site-packages/twisted/internet/defer.py:398: FutureWarning: h
ex()/oct() of negative int will return a signed string in Python 2.4 and up
  return "<%s at %s>" % (cname, hex(id(self)))
[PASS]       2005-03-27T21:45:20Z   pvmd-path.apt
[PASS]       2005-03-27T21:45:20Z   envvar-pvmroot.apt
[PASS]       2005-03-27T21:45:20Z   envvar-pvmarch.apt

...Hit <ENTER> key to exit...
```

Figure 32: Test the cluster.

E.11 Delete Node Button

The Delete Node button can be used to delete clients from an OSCAR cluster. Note that this button only deletes OSCAR's knowledge of the clients – what physically happens to that client is not OSCAR's concern. This example shows deleting one of the clients setup in the previous sections – `oscarnode2`. The next section (Section E.12) will show adding it back. See the details on deleting clients in Section 5.14.2, page 31.


Delete Oscar Nodes

In order to delete OSCAR clients from your cluster, select the nodes you wish to delete and press the Delete Clients button.

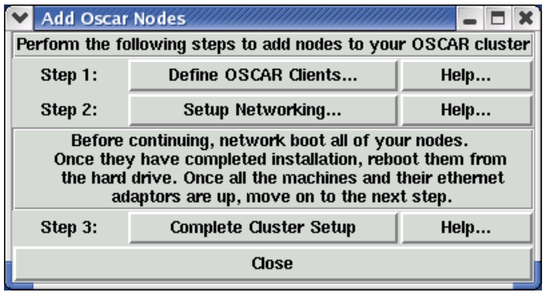
oscarnode1
oscarnode2

Clients deleted.

Figure 33: Delete OSCAR clients. First image is the initial window, second image is with a client selected, and third image is when the action has completed.

E.12 Add Node Button

The Add Node button will add clients into an existing OSCAR cluster. In this example, we will add back `oscarnode2` into the cluster. See the details of adding a client in Section 5.14.1, page 30.


Add Oscar Nodes

Perform the following steps to add nodes to your OSCAR cluster

Step 1: Define OSCAR Clients... Help...

Step 2: Setup Networking... Help...

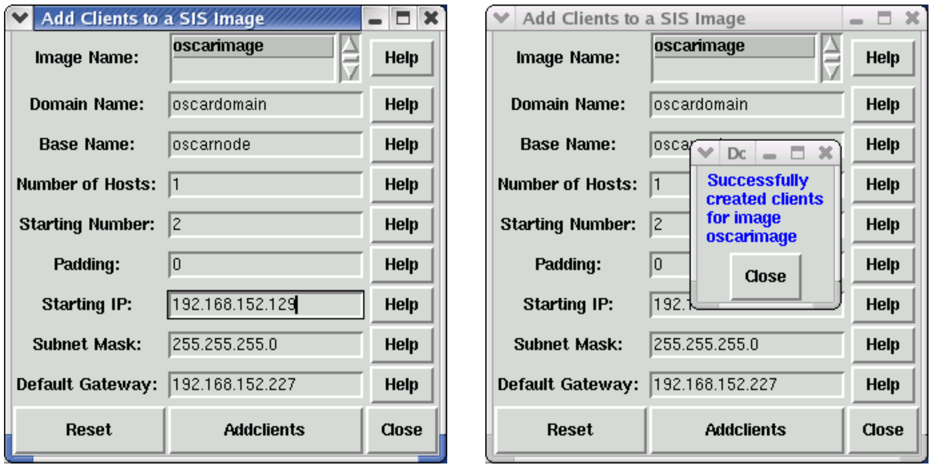
Before continuing, network boot all of your nodes. Once they have completed installation, reboot them from the hard drive. Once all the machines and their ethernet adaptors are up, move on to the next step.

Step 3: Complete Cluster Setup Help...

Figure 34: Add OSCAR Clients.

E.13 Install/Uninstall OSCAR Packages

The Install/Uninstall OSCAR Packages button will let you add/remove packages from and an OSCAR cluster after the initial installation. See the details and more in-depth look at the underlying concepts of this new feature in Section 5.15, page 31.



Add Clients to a SIS Image

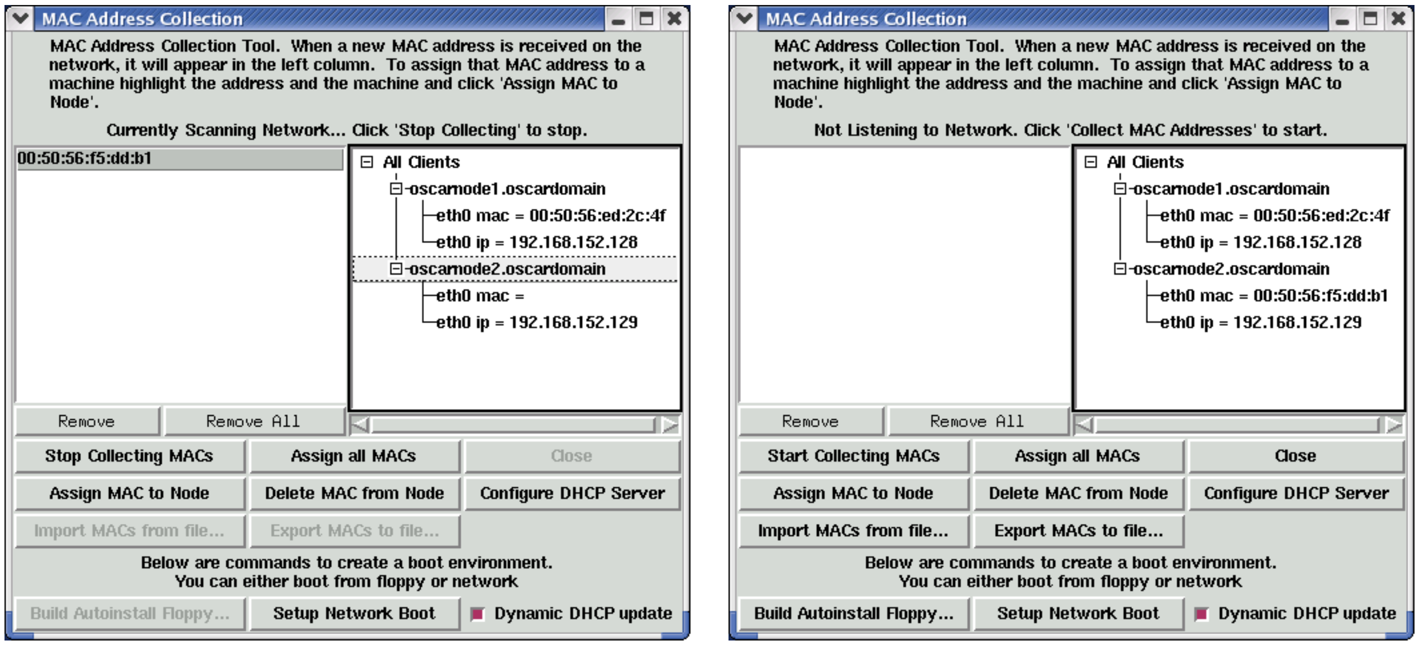
Image Name:	oscarimage	Help
Domain Name:	oscardomain	Help
Base Name:	oscarnode	Help
Number of Hosts:	1	Help
Starting Number:	2	Help
Padding:	0	Help
Starting IP:	192.168.152.129	Help
Subnet Mask:	255.255.255.0	Help
Default Gateway:	192.168.152.227	Help

Reset | Addclients | Close

Successfully created clients for image oscarimage

Close

Figure 35: Add node / defining the clients.



MAC Address Collection

MAC Address Collection Tool. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click 'Assign MAC to Node'.

Currently Scanning Network... Click 'Stop Collecting' to stop.

00:50:56:f5:dd:b1

```
⊟ All Clients
  ⊟-oscarnode1.oscardomain
     ├eth0 mac = 00:50:56:ed:2c:4f
     └eth0 ip = 192.168.152.128
  ⊟-oscarnode2.oscardomain
     ├eth0 mac =
     └eth0 ip = 192.168.152.129
```

Remove | Remove All

Stop Collecting MACs | Assign all MACs | Close

Assign MAC to Node | Delete MAC from Node | Configure DHCP Server

Import MACs from file... | Export MACs to file...

Below are commands to create a boot environment.
You can either boot from floppy or network

Build Autoinstall Floppy... | Setup Network Boot | Dynamic DHCP update

MAC Address Collection

MAC Address Collection Tool. When a new MAC address is received on the network, it will appear in the left column. To assign that MAC address to a machine highlight the address and the machine and click 'Assign MAC to Node'.

Not Listening to Network. Click 'Collect MAC Addresses' to start.

```
⊟ All Clients
  ⊟-oscarnode1.oscardomain
     ├eth0 mac = 00:50:56:ed:2c:4f
     └eth0 ip = 192.168.152.128
  ⊟-oscarnode2.oscardomain
     ├eth0 mac = 00:50:56:f5:dd:b1
     └eth0 ip = 192.168.152.129
```

Remove | Remove All

Start Collecting MACs | Assign all MACs | Close

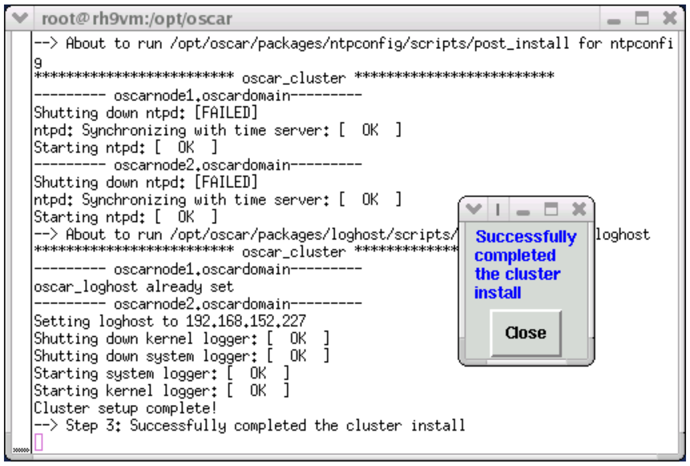
Assign MAC to Node | Delete MAC from Node | Configure DHCP Server

Import MACs from file... | Export MACs to file...

Below are commands to create a boot environment.
You can either boot from floppy or network

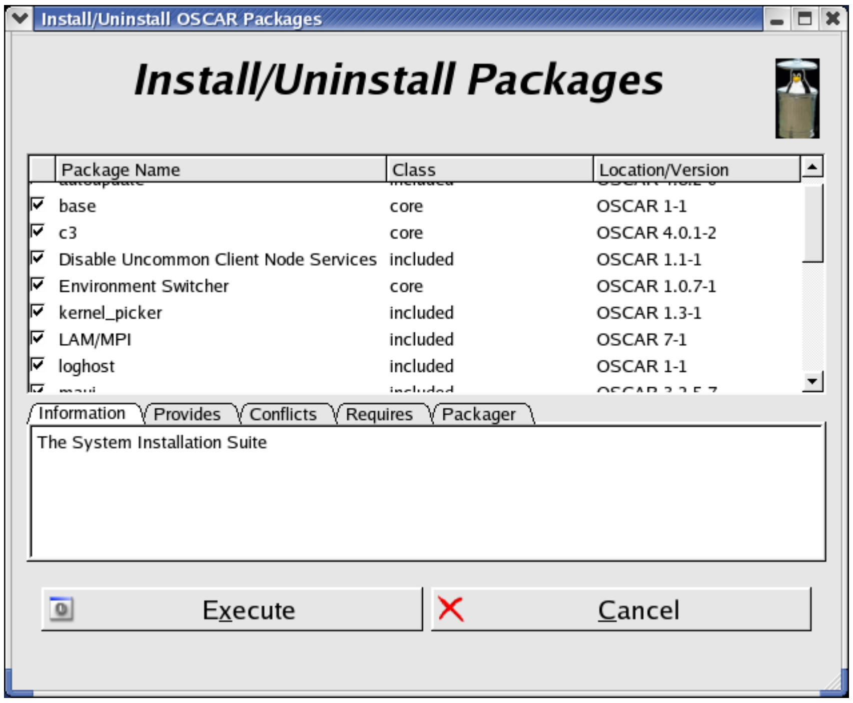
Build Autoinstall Floppy... | Setup Network Boot | Dynamic DHCP update

Figure 36: Add node / setup networking.



```
--> About to run /opt/oscar/packages/ntpconfig/scripts/post_install for ntpconfi
g
************************ oscar_cluster ************************
--------- oscarnode1.oscardomain---------
Shutting down ntpd: [FAILED]
ntpd: Synchronizing with time server: [  OK  ]
Starting ntpd: [  OK  ]
--------- oscarnode2.oscardomain---------
Shutting down ntpd: [FAILED]
ntpd: Synchronizing with time server: [  OK  ]
Starting ntpd: [  OK  ]
--> About to run /opt/oscar/packages/loghost/scripts/                loghost
************************ oscar_cluster *************
--------- oscarnode1.oscardomain---------
oscar_loghost already set
--------- oscarnode2.oscardomain---------
Setting loghost to 192.168.152.227
Shutting down kernel logger: [  OK  ]
Shutting down system logger: [  OK  ]
Starting system logger: [  OK  ]
Starting kernel logger: [  OK  ]
Cluster setup complete!
--> Step 3: Successfully completed the cluster install
```

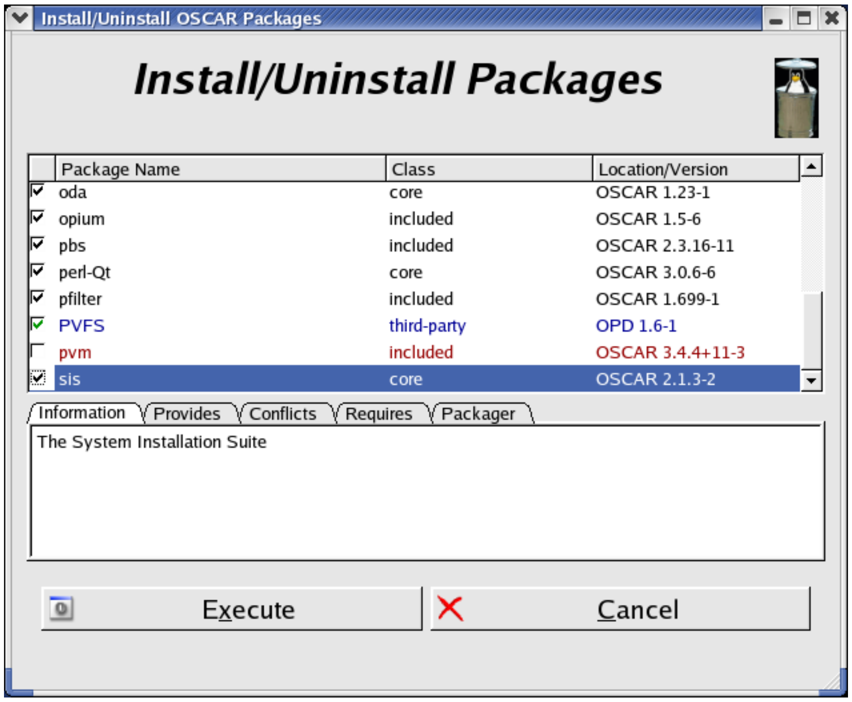
Figure 37: Add node / complete cluster setup.



	Package Name	Class	Location/Version
☑	base	core	OSCAR 1-1
☑	c3	core	OSCAR 4.0.1-2
☑	Disable Uncommon Client Node Services	included	OSCAR 1.1-1
☑	Environment Switcher	core	OSCAR 1.0.7-1
☑	kernel_picker	included	OSCAR 1.3-1
☑	LAM/MPI	included	OSCAR 7-1
☑	loghost	included	OSCAR 1-1

The System Installation Suite

Figure 38: Install/Uninstall packages from an existing OSCAR cluster.



	Package Name	Class	Location/Version
✓	oda	core	OSCAR 1.23-1
✓	opium	included	OSCAR 1.5-6
✓	pbs	included	OSCAR 2.3.16-11
✓	perl-Qt	core	OSCAR 3.0.6-6
✓	pfilter	included	OSCAR 1.699-1
✓	PVFS	third-party	OPD 1.6-1
	pvm	included	OSCAR 3.4.4+11-3
✓	sis	core	OSCAR 2.1.3-2

The System Installation Suite

Figure 39: Individual OSCAR packages are color-coded to show their current installation state.