

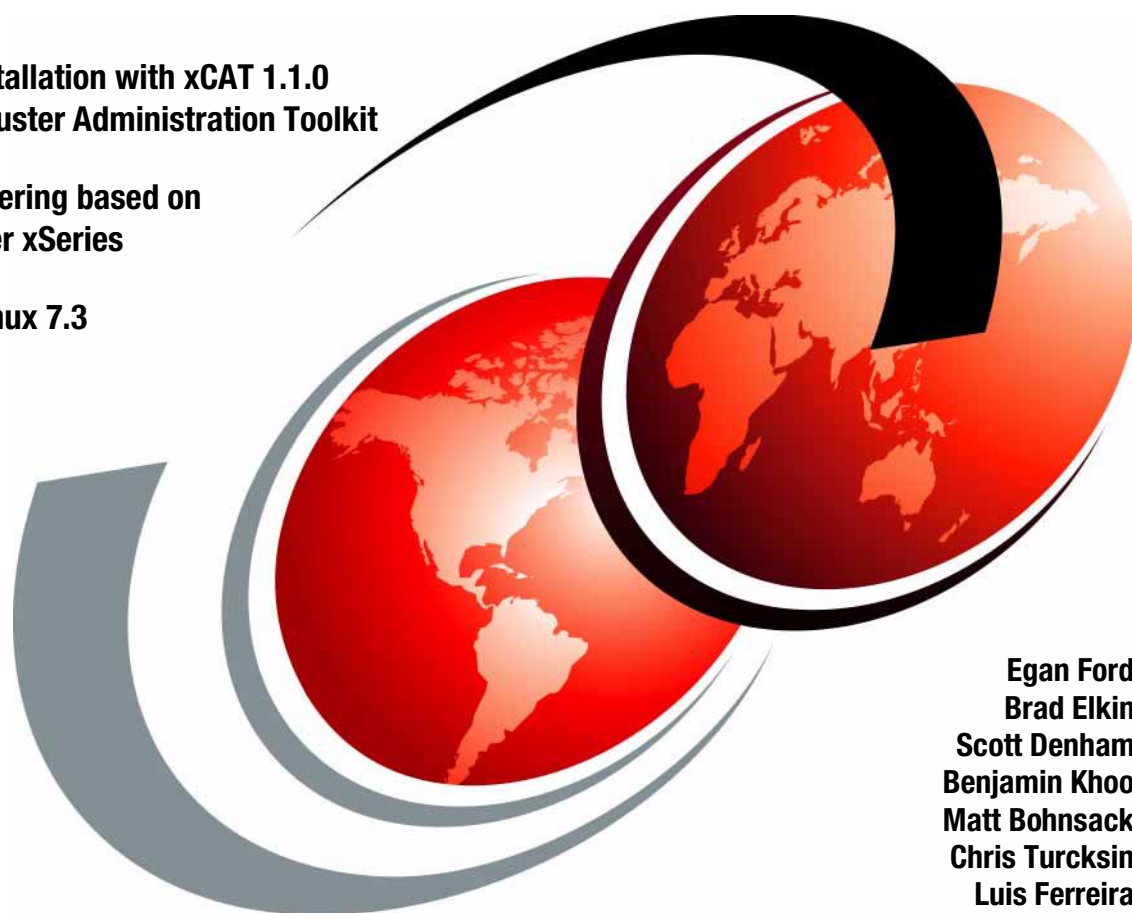


# Building a Linux HPC Cluster with xCAT

Cluster installation with xCAT 1.1.0  
Extreme Cluster Administration Toolkit

Linux clustering based on  
IBM eServer xSeries

Red Hat Linux 7.3



Egan Ford  
Brad Elkin  
Scott Denham  
Benjamin Khoo  
Matt Bohnsack  
Chris Turcksin  
Luis Ferreira





International Technical Support Organization

**Building a Linux HPC Cluster with xCAT**

September 2002

**Note:** Before using this information and the product it supports, read the information in “Notices” on page xvii.

### **First Edition (September 2002)**

This edition applies to Red Hat® Linux® Version 7.3 for Intel® Architecture.

© Copyright International Business Machines Corporation 2002. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



# Contents

<b>Figures</b> .....	xiii
<b>Tables</b> .....	xv
<b>Notices</b> .....	xvii
Trademarks .....	xviii
<b>Preface</b> .....	xxi
The team that wrote this redbook .....	xxi
Acknowledgements .....	xxiii
Become a published author .....	xxv
Comments welcome .....	xxv
<b>Chapter 1. HPC clustering concepts</b> .....	1
1.1 What a cluster is .....	2
1.1.1 High-Performance Computing cluster .....	2
1.1.2 Beowulf clusters .....	3
1.2 IBM Linux clusters .....	4
1.2.1 xSeries custom-order cluster .....	4
1.2.2 IBM eServer Cluster 1300 .....	5
1.2.3 The new IBM eServer Cluster 1350 .....	6
1.3 Making up an HPC cluster .....	7
1.3.1 Logical functions that a node can provide .....	7
1.3.2 xSeries models used in our cluster .....	10
1.3.3 Other cluster components .....	12
1.4 Software .....	15
1.4.1 IBM Cluster Systems Management for Linux .....	15
<b>Chapter 2. xCAT introduction</b> .....	17
2.1 What xCAT is .....	19
2.1.1 Download xCAT .....	20
2.1.2 Directory structure .....	20
2.2 Installing a Linux cluster with xCAT .....	22
2.2.1 Planning .....	22
2.2.2 Hardware preparation .....	26
2.2.3 Management node installation .....	26
2.2.4 Cluster installation .....	27
<b>Chapter 3. Hardware preparation</b> .....	31

3.1 Node hardware installation . . . . .	32
3.2 Populating the rack and cabling . . . . .	33
3.3 Cables in our cluster . . . . .	40
<b>Chapter 4. Management node installation . . . . .</b>	<b>43</b>
4.1 Resources to install Red Hat Linux . . . . .	44
4.2 Red Hat installation steps . . . . .	45
4.3 Post-installation steps . . . . .	50
4.3.1 Copy Red Hat install CD-ROMs . . . . .	50
4.3.2 Install Red Hat errata . . . . .	51
4.3.3 Updating third party drivers . . . . .	54
<b>Chapter 5. Management node configuration . . . . .</b>	<b>57</b>
5.1 Install xCAT . . . . .	58
5.2 Populate tables . . . . .	58
5.2.1 Site definition. . . . .	60
5.2.2 Hosts file . . . . .	61
5.2.3 List of nodes and groups. . . . .	63
5.2.4 Installation resources . . . . .	64
5.2.5 Node types . . . . .	65
5.2.6 Node hardware management . . . . .	65
5.2.7 MPN topology . . . . .	66
5.2.8 MPA configuration. . . . .	67
5.2.9 Power control with APC MasterSwitch . . . . .	68
5.2.10 MAC address collection using Cisco 3500-series . . . . .	68
5.2.11 Console server configuration . . . . .	69
5.2.12 Password table . . . . .	71
5.3 Configure management node services . . . . .	71
5.3.1 Turn off services you do not want . . . . .	71
5.3.2 Configure system logging . . . . .	72
5.3.3 Configure SNMP . . . . .	73
5.3.4 Configure TFTP. . . . .	74
5.3.5 Configure NFS . . . . .	74
5.3.6 Configure NTP . . . . .	75
5.3.7 Configure SSH . . . . .	76
5.3.8 Configure the console server . . . . .	77
5.3.9 Configure DNS . . . . .	77
5.3.10 Configure DHCP . . . . .	78
5.4 Final preparation . . . . .	79
5.4.1 Prepare the boot files for stages 2 and 3 . . . . .	79
5.4.2 Prepare the Kickstart files . . . . .	80
5.4.3 Prepare the post installation directory structure . . . . .	80
<b>Chapter 6. Cluster installation . . . . .</b>	<b>83</b>

6.1 Stage 1: Hardware setup . . . . .	84
6.1.1 Network switch setup . . . . .	84
6.1.2 Management Processor Adapter setup . . . . .	91
6.1.3 Terminal server setup . . . . .	93
6.1.4 APC MasterSwitch setup . . . . .	96
6.1.5 BIOS and firmware updates . . . . .	97
6.2 Stage 2: MAC address collection . . . . .	100
6.3 Stage 3: Management processor setup . . . . .	103
6.4 Stage 4: Node installation . . . . .	107
6.4.1 Creating a template file . . . . .	107
6.4.2 Creating a custom kernel RPM image . . . . .	109
6.4.3 Creating a custom kernel tarball image . . . . .	109
6.4.4 Installing the nodes . . . . .	110
6.4.5 Post-installation . . . . .	114
<b>Appendix A. xCAT commands . . . . .</b>	<b>117</b>
Command reference . . . . .	118
addclusteruser - Add a cluster user . . . . .	120
Options . . . . .	121
Files . . . . .	121
Diagnostics . . . . .	121
Examples . . . . .	121
Bugs . . . . .	122
Author . . . . .	122
mpacheck - Check MPA and MPA settings . . . . .	123
Synopsis . . . . .	123
Description . . . . .	123
Options . . . . .	123
Files . . . . .	123
Diagnostics . . . . .	123
Examples . . . . .	124
Bugs . . . . .	124
Author . . . . .	125
See also . . . . .	125
mpareset - Reset MPAs . . . . .	126
Synopsis . . . . .	126
Description . . . . .	126
Options . . . . .	126
Files . . . . .	126
Diagnostics . . . . .	126
Examples . . . . .	127
Bugs . . . . .	127
Author . . . . .	127

See also .....	127
mpascan - Scan MPA for RS485 chained nodes .....	128
Synopsis .....	128
Description .....	128
Options .....	128
Files .....	128
Diagnostics .....	128
Examples .....	129
Bugs .....	129
Author .....	129
See also .....	129
mpasetup - Set MPA settings .....	130
Synopsis .....	130
Description .....	130
Options .....	130
Files .....	130
Diagnostics .....	130
Examples .....	131
Author .....	132
Bugs .....	132
See also .....	132
nodels - List node properties from tables .....	133
Synopsis .....	133
Description .....	133
Options .....	133
Author .....	133
noderange - Generate a list of node names .....	134
Synopsis .....	134
Description .....	134
Options .....	137
Environmental variables .....	137
Files .....	138
Example .....	138
Bugs/features .....	139
Author .....	139
nodeset - Set the boot state for a noderange .....	140
Synopsis .....	140
Description .....	140
Options .....	140
Files .....	141
Diagnostics .....	142
Examples .....	143
Bugs .....	143

Author .....	143
See also .....	144
pping - Parallel ping .....	145
Synopsis .....	145
Description .....	145
Options .....	145
Files .....	145
Diagnostics .....	145
Examples .....	145
Bugs .....	146
Author .....	146
See also .....	146
prcp - Parallel remote copy .....	147
Synopsis .....	147
Description .....	147
Options .....	147
Files .....	147
Diagnostics .....	148
Examples .....	148
Bugs .....	148
Author .....	148
See also .....	148
prsync - parallel rsync .....	149
Synopsis .....	149
Description .....	149
Options .....	149
Files .....	149
Diagnostics .....	149
Examples .....	150
Bugs .....	150
Author .....	150
See also .....	150
psh - Parallel remote shell .....	151
Synopsis .....	151
Description .....	151
Options .....	151
Files .....	151
Diagnostics .....	152
Examples .....	152
Bugs .....	152
Author .....	152
See also .....	152
rcons - remote console .....	153

Synopsis .....	153
Description .....	153
Options .....	153
Files .....	153
Diagnostics .....	153
Examples .....	154
Bugs .....	154
Author .....	154
See also .....	154
reventlog - Retrieve or clear remote hardware event logs .....	155
Synopsis .....	155
Description .....	155
Options .....	155
Files .....	155
Diagnostics .....	155
Examples .....	156
Bugs .....	157
Author .....	157
See also .....	157
rinstall - Remote network install .....	158
Synopsis .....	158
Description .....	158
Options .....	158
Files .....	158
Diagnostics .....	158
Examples .....	158
Bugs .....	159
Author .....	159
See also .....	159
rinv - Remote hardware inventory .....	160
Synopsis .....	160
Description .....	160
Options .....	160
Files .....	160
Diagnostics .....	161
Examples .....	161
Bugs .....	162
Author .....	162
See also .....	162
rpower - Remote power control .....	163
Synopsis .....	163
Description .....	163
Options .....	163

Files . . . . .	163
Diagnostics . . . . .	163
Examples . . . . .	164
Bugs . . . . .	164
Author . . . . .	164
See also . . . . .	165
rreset - Remote hard reset . . . . .	166
Synopsis . . . . .	166
Description . . . . .	166
Options . . . . .	166
Files . . . . .	166
Diagnostics . . . . .	166
Examples . . . . .	167
Bugs . . . . .	167
Author . . . . .	167
See also . . . . .	167
rvid - Remote video (VGA) . . . . .	168
Synopsis . . . . .	168
Description . . . . .	168
Options . . . . .	168
Files . . . . .	168
Diagnostics . . . . .	169
Examples . . . . .	169
Bugs . . . . .	170
Author . . . . .	170
See also . . . . .	170
rvitals - Remote hardware vitals . . . . .	171
Synopsis . . . . .	171
Description . . . . .	171
Options . . . . .	171
Files . . . . .	171
Diagnostics . . . . .	172
Examples . . . . .	173
Bugs . . . . .	173
Author . . . . .	173
See also . . . . .	173
wcons - Windowed remote console . . . . .	174
Synopsis . . . . .	174
Description . . . . .	174
Options . . . . .	174
Files . . . . .	175
Diagnostics . . . . .	175
Examples . . . . .	175

Bugs	176
Author	176
See also	176
winstall - Windowed remote network install	177
Synopsis	177
Description	177
Options	177
Files	178
Diagnostics	178
Examples	178
Bugs	179
Author	179
See also	179
wkill - Windowed remote console kill	180
Synopsis	180
Description	180
Options	180
Files	180
Diagnostics	180
Examples	180
Bugs	181
Author	181
See also	181
wvid - Windowed remote video (VGA)	182
Synopsis	182
Description	182
Options	182
Files	183
Diagnostics	183
Example	184
Bugs	184
Author	184
See also	184
<b>Appendix B. xCAT configuration tables</b>	185
site.tab	188
nodelist.tab	193
noderes.tab	194
nodetype.tab	196
nodehm.tab	197
mpa.tab	201
apc.tab	202
apcp.tab	203



mac.tab . . . . .	204
cisco3500.tab . . . . .	205
passwd.tab . . . . .	206
conserver.tab . . . . .	208
rtel.tab . . . . .	209
tty.tab . . . . .	210
<b>Appendix C. Other hardware components . . . . .</b>	<b>211</b>
IBM Advanced Systems Management Adapter . . . . .	212
Equinox ESP Terminal Servers . . . . .	212
iTouch Communications IR-8000 Terminal Servers. . . . .	217
Myrinet . . . . .	218
Myrinet switch layout. . . . .	219
Setting up the Myrinet switch . . . . .	221
Installing the Myrinet software. . . . .	222
<b>Appendix D. Application examples . . . . .</b>	<b>225</b>
User accounts . . . . .	226
MPICH . . . . .	226
Persistence of Vision Raytracer (POVray) . . . . .	228
Serial POVray . . . . .	228
Distributed POVray using MPI-POVray. . . . .	230
High Performance Linpack (HPL). . . . .	232
Installing ATLAS . . . . .	233
Installing HPL . . . . .	233
<b>Related publications . . . . .</b>	<b>237</b>
IBM Redbooks . . . . .	237
Other resources . . . . .	237
Referenced Web sites . . . . .	237
How to get IBM Redbooks . . . . .	240
IBM Redbooks collections. . . . .	241
<b>Glossary . . . . .</b>	<b>243</b>
<b>Index . . . . .</b>	<b>245</b>



# Figures

0-1	The Blue Tuxedo Team . . . . .	xxiii
1-1	High-Performance Computing cluster . . . . .	3
1-2	Beowulf logical view . . . . .	4
1-3	Logical structure of a cluster . . . . .	8
1-4	Model 342 management node . . . . .	11
1-5	Model 330 for compute nodes . . . . .	12
1-6	Cable chain technology . . . . .	14
1-7	Management processor network . . . . .	15
2-1	IP address octets . . . . .	23
2-2	Network boot and installation process . . . . .	30
3-1	x330 with PCI cards installed . . . . .	33
3-2	MPN and C2T cabling . . . . .	35
3-3	Terminal server cables (left) and FastEthernet cabling (right) . . . . .	36
3-4	Power distribution units . . . . .	38
3-5	Cluster Ethernet, MPN, and C2T cabling . . . . .	39
3-6	Cables on our master node (x342) . . . . .	40
3-7	Cables on our compute nodes (x330) . . . . .	41
4-1	xSeries 342 support . . . . .	44
4-2	IBM @server xSeries 342 - Installing Linux . . . . .	45
6-1	Installation screens . . . . .	111
A-1	Windowed remote console . . . . .	176
A-2	Windowed remote network install . . . . .	179
A-3	Windowed remote video (VGA) . . . . .	184
C-1	Myrinet - Single switch layout . . . . .	219
C-2	Myrinet - Tree switch layout . . . . .	220
C-3	Myrinet - Polygon switch layout . . . . .	221



# Tables

1-1	Typical Linux cluster	10
2-1	Naming convention	22
2-2	IP address assignments	23
2-3	VLAN assignments	25
5-1	xCAT configuration tables overview	59
A-1	xCAT commands	118
A-2	Site.tab fields for addclusteruser	120
A-3	addclusteruser prompts	121
B-1	xCAT tables description	185
B-2	Definition of site.tab parameters	188
B-3	Definition of nodelist.tab parameters	193
B-4	Definition of noderes.tab parameters	194
B-5	Definition of nodetype.tab parameters	196
B-6	Definition of nodehm.tab parameters	197
B-7	Definition of mpa.tab parameters	201
B-8	Definition of apc.tab parameters	202
B-9	Definition of apcp.tab parameters	203
B-10	Definition of mac.tab parameters	204
B-11	Definition of cisco3500.tab parameters	205
B-12	Definition of passwd.tab parameters	206
B-13	Definition of conserver.tab parameters	208
B-14	Definition of rtel.tab parameters	209
B-15	Definition of tty.tab parameters	210



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.





Maui Scheduler is a trademark of Science & Technology Corporation @ UNM. Software developed for The University of New Mexico.

Other company, product, and service names may be trademarks or service marks of others.



# Preface

This redbook describes how to implement Linux cluster on IBM eServer xSeries hardware using the Extreme Cluster Administration Toolkit, known as xCAT, and other third-party software. It covers xCAT Version 1.1.0 running on Linux Red Hat 7.3. This book guides system architects and systems engineers through a basic understanding of cluster technology, terminology, and Linux High-Performance Computing (HPC) clusters. Also, it teaches you the installation process.

Management tools are provided to easily manage a large number of compute nodes that use the built-in features of Linux and the advanced management capabilities of the IBM eServer xSeries Management Processor Network.

## The team that wrote this redbook

This redbook was produced by the Blue Tuxedo Team, a team of specialists from around the world working at the International Technical Support Organization, Austin Center.

**Luis Ferreira** (also known as “Luix”) is a Software Engineer at IBM Corporation - International Technical Support Organization, Austin Center, working on Linux and AIX projects. He has 18 years of experience with UNIX-like operating systems, and holds a MSc. Degree in System Engineering from Universidade Federal do Rio de Janeiro in Brazil. Before joining the ITSO, Luis worked at Tivoli Systems as a Certified Tivoli Consultant, at IBM Brasil as a Certified IT Specialist, and at Cobra Computadores as a Kernel Developer and Software Designer. His e-mail address is [luix@us.ibm.com](mailto:luix@us.ibm.com).

**Christopher Turcksin** (also known as “Wabbit”) is an IT Specialist at IBM Global Services at the Scottish Service Centre in Greenock, Scotland. He has eight years of experience with Linux and has currently been working with xCAT and IBM Linux clusters. Before joining the Scottish Service Centre, Christopher worked as a Software Developer (writing code in C, C++, and Java) and a System Support Analyst supporting customers and business partners at the IBM EMEA HelpCentre. His e-mail address is [turcksin@uk.ibm.com](mailto:turcksin@uk.ibm.com).

**Brad Elkin** is a Senior Software Engineer in Minnesota, USA. He has 15 years of experience in High-Performance Computing. He has worked in the Life Science Technical Solutions Development Group in IBM for a year. His areas of expertise include Computational Chemistry, Bioinformatics, and Computational Fluid

Dynamics. Brad has a Ph.D. in Chemical Engineering from the University of Pennsylvania. His e-mail is [be@us.ibm.com](mailto:be@us.ibm.com).

**Scott Denham** is an IT Architect at the IBM Industrial Sector Center of Competency in Houston, Texas. He majored in Electrical Engineering at the University of Houston, and worked for 28 years in the petroleum exploration industry on High-Performance Computing and Seismic Software Applications Development before joining IBM in 2000. Scott's current responsibility includes pre-sales technical support and performance evaluation for pSeries and xSeries HPC customers. His areas of expertise include I/O programming, array processors, AIX and the RS/6000 SP system, high-performance network configuration, and Linux clusters. Scott has been working with xCAT clusters in petroleum since January, 2001. His e-mail address is [sdenham@us.ibm.com](mailto:sdenham@us.ibm.com).

**Benjamin Khoo** is an IT Specialist in IBM Global Services Singapore. He majored in Electrical and Electronics Engineering at the National University of Singapore. He had three years of HPC experience before joining IBM. His areas of responsibility includes Linux, Linux High Performance and High Availability Clusters, and recently, Grid Computing. His e-mail address is [khoob@sg.ibm.com](mailto:khoob@sg.ibm.com).

**Matt Bohnsack** is a Linux Cluster Architect for IBM Global Services. He has implemented over 30 Linux clusters based on xCAT and is the creator and maintainer of the <http://x-cat.org> Web site. He has been working with Linux since 1994 and holds a B.S. in Electrical Engineering from Iowa State University. His e-mail address is [bohnsack@us.ibm.com](mailto:bohnsack@us.ibm.com).

**Egan Ford** is a Linux Cluster Architect for IBM Advance Technical Support. He has 14 years of UNIX/Linux experience and three years with Linux HPC clusters. Egan was one of the pioneers of Linux HPC clusters at IBM and wrote xCAT to fulfill the needs of IBM Linux HPC customers. His e-mail address is [egan@us.ibm.com](mailto:egan@us.ibm.com).



Figure 0-1 The Blue Tuxedo Team

## Acknowledgements

Figure 0-1 shows the Blue Tuxedo Team. From left to right they are Brad, Christopher, Scott, Benjamin, Luis, Matt, and Egan.

This redbook was produced based on xCAT, which was designed and written by Egan Ford and also based on the following Redbooks:

- ▶ *Linux HPC Cluster Installation*, SG24-6041, written by Gregory Kettmann, Andreas Thomasch, Eileen Silcocks, Jacob Chen, Jean-Claude Daunois, Jens Ihamo, Makoto Harada, Steve Hill, Walter Bernocchi, Egan Ford, and Luis Ferreira.
- ▶ *Linux Clustering with CSM and GPFS*, SG24-6601, written by Jean-Claude Daunois, Eric Monjoin, Antonio Forster, Bart Jacob, and Luis Ferreira.

Thanks to the following people for their contributions to this project:

Lupe Brown, Bart Jacob, Wade Wallace, Julie Czubik, and Chris Blatchley  
**International Technical Support Organization, Austin Center**

Nina (and Anishka) Wilner  
**pSeries Technical Solution Manager LifeSciences, IBM Austin**

Gabriel Sallah and David McLaughlin  
**IBM Greenock, Scotland**

Merlin Glynn, Dan O Cummings, Tonko De Rooy, Scott Hanson, and Wes Kinard  
**ATS Linux Cluster Team, IBM Dallas, USA**

Rebecca Austen  
**Linux Cluster Marketing, IBM Austin, USA**

Bruce Potter  
**Linux Cluster Architect, IBM Poughkeepsie, USA**

Joseph Banas  
**eServer Linux Clusters, IBM Poughkeepsie, USA**

Sharon Dobbs  
**Server Group, IBM Austin, USA**

Andreas Hermelink  
**Worldwide Linux Technical Sales Support, IBM Somers, USA**

Steve Hill  
**High-Performance Computing, IBM Hursley, England**

Walter Bernocchi  
**EMEA South Region Linux Impact Team, IBM Milan, Italy**

Rainer Kubesch  
**IBM Berlin, Germany**

Rohit Bhargav  
**Linux Practices ITS Asia-Pacific, IBM Global Services, India**

Eileen Silcocks  
**EMEA Technical Support Education, IBM Greenock, Scotland**

Cameron Ferstat  
**Consulting IT Specialist, IBM Sydney, Australia**

Joe Vaught and Doug Huckaby  
**PCPC Inc., Houston, USA**

Special Thanks to Alan Fishman and Peter Nielsen (Solution Managers, IGS Linux Services), and Joanne Luedtke (International Technical Support Organization Manager, Austin Center) for their effort and support for this project.

## Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ Send your comments in an Internet note to:

[redbook@us.ibm.com](mailto:redbook@us.ibm.com)

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization  
Dept. JN9B Building 003 Internal Zip 2834  
11400 Burnet Road  
Austin, Texas 78758-3493







# HPC clustering concepts

This chapter introduces the High-Performance Computing clustering concepts and terminology that are used throughout the rest of this book. We also discuss and describe some common components that make up generic clusters.

These are the discussed topics:

- ▶ What a High-Performance Computing cluster is
- ▶ Cluster nodes: Types, functions, and models
- ▶ Other cluster components, such as, networking, terminal server and management processor

The redbook assumes that the reader is knowledgeable of advanced Linux skills, such as installation, configuration, and management.

## 1.1 What a cluster is

In its simplest form, a cluster is two or more computers that work together to provide a solution. This should not be confused with a more common client-server model of computing where an application may be logically divided such that one or more clients request services of one or more servers. The idea behind clusters is to join the computing powers of the nodes involved to provide higher scalability, more combined computing power, or to build in redundancy to provide higher availability. So rather than a simple client making requests of one or more servers, clusters utilize multiple machines to provide a more powerful computing environment through a single system image.

### 1.1.1 High-Performance Computing cluster

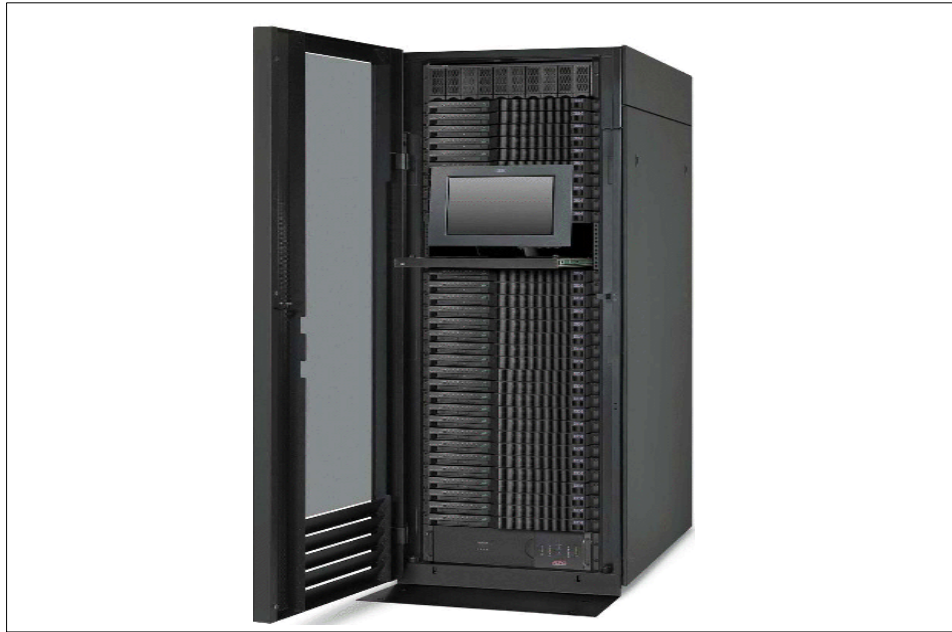
High-Performance Computing clusters are designed to use parallel computing to apply more processor power for the solution of a problem. There are many examples of scientific computing using multiple low-cost processors in parallel to perform large numbers of operations. This is referred to as parallel computing or parallelism. Thomas Sterling, in his paper entitled *How to Build a Beowulf*, stated “Parallelism is the ability of many independent threads of control to make progress simultaneously toward the completion of a task.”

A High-Performance cluster, as seen on Figure 1-1 on page 3, is typically made up of a large number of nodes. Clusters of hundreds of nodes are not uncommon. Creating an architecture for this kind of cluster brings its own challenges, which includes:

- ▶ How to install and maintain the operating system and the application environment on all nodes
- ▶ How to pro-actively manage these nodes issuing commands as well as gracefully handling failures
- ▶ The requirement for parallel, concurrent, and high-performance access to the same file system
- ▶ Inter-process communication between the nodes to coordinate the work that must be done in parallel

The goal is to provide the image of a single system by managing, operating, and coordinating a large number of discrete computers.

Often in this environment, a user interacts with a specific node to initiate or schedule a job to be run. The application, in conjunction with various functions within the cluster, then determines how this job is spread across the various nodes of the cluster to take advantage of the resources available to produce the desired result.



*Figure 1-1 High-Performance Computing cluster*

## **1.1.2 Beowulf clusters**

Beowulf is mainly based on commodity hardware, software, and standards. It is one of the architectures used when intensive computing applications are essential for a successful result. It is a union of several components that, if tuned and selected appropriately, can speed up the execution of a well-written application. A logical view of Beowulf architecture is illustrated in Figure 1-2 on page 4.

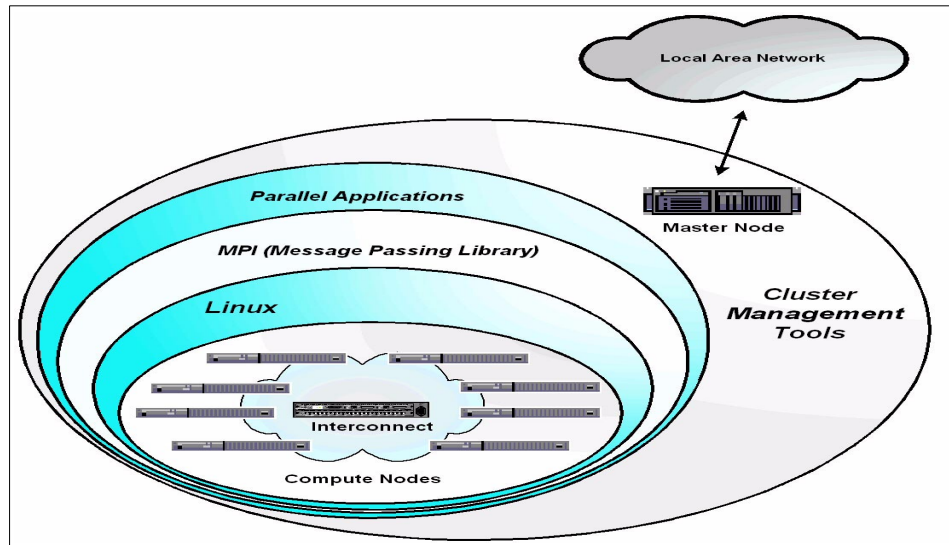


Figure 1-2 Beowulf logical view

## 1.2 IBM Linux clusters

Today's e-infrastructure requires IT systems to meet increasing demands, while offering the flexibility and manageability to rapidly develop and deploy new services. IBM Linux clusters address all these customer needs by providing hardware and software solutions to satisfy the IT requirements.

### 1.2.1 xSeries custom-order cluster

Clustered computing has been with IBM for several years. IBM, through its services arm (IBM Global Services), has been involved in helping customers create Linux-based clusters. Because Linux clustering is a relatively recent phenomenon, there has not been a set of best practices or any standard cluster configuration that customers could order off-the-shelf. In most cases, each customer had to “reinvent the wheel” when designing and procuring all the components for a cluster. However, based on the IGS experience, many of these best practices have been developed and practical experience has been built while creating Linux-based clusters in a variety of environments. Based on this experience, IBM offers solutions that combine these experiences, best practices, and the most commonly used software and hardware components to provide a cluster offering that can be deployed quickly in a variety of environments.

## 1.2.2 IBM eServer Cluster 1300

The IBM eServer Cluster 1300 is a solution that provides a pre-packaged set of hardware, software, and services, which allows customers to quickly deploy cluster-based solutions. Though Linux clusters have been growing in popularity, most deployments have often taken months or longer before all of the hardware and software components could be obtained and put in place to form a production environment.

The IBM eServer Cluster 1300 consists of a combination of IBM and non-IBM hardware and software that can be configured to meet the specific needs of a particular customer. This configuration occurs before the cluster is delivered to the customer. That is, what is delivered to the customer is one or more racks with the hardware and software already installed, configured, and tested. Once onsite, only minor customer-specific configuration tasks need to be performed. IBM provides services to perform these as part of the product offering.

Based on the specifics of the application(s) that the customer provides to run on these clusters, an IBM eServer Cluster 1300 can literally be up and in production in just a matter of days after the system arrives.

IBM's Linux-based cluster offering brings together the hardware, software, and services required for a complete cluster deployment. Because of the scalability, you can configure a cluster to meet your current needs and expand it as your business changes.

This cluster is built on Intel architecture, rack-optimized servers. Each server can be configured to match the requirements of the applications that it will run.

The IBM eServer Cluster 1300 is ordered as an integrated offering. Therefore, instead of having to develop a system design and then obtain and integrate all of the individual components, the entire solution can be delivered as a unit. IBM provides tools to easily configure and order a cluster, thereby speeding its actual deployment.

In addition, IBM provides end-to-end support for *all* cluster components, including industry-leading technologies from OEM suppliers such as Myricom and Cisco.

A Linux cluster utilizes the Linux operating system on each of the nodes of the cluster. However, the combination of hardware and Linux running on each node does not necessarily provide an operational cluster solution. There must be cluster-specific management added to the mix to enable the cluster to act as a single system. This management software is IBM Cluster Systems Management (CSM) for Linux.

In addition, IBM General Parallel File System (GPFS) for Linux can also be utilized in this solution to provide high speed and reliable storage access from large numbers of nodes within the cluster.

For more information about hardware, software, and service components that make up the IBM @server Cluster 1300 product offering, refer to the redbook *Linux Clustering with CSM and GPFS*, GS24-6601, and the IBM @server Cluster Web site at:

<http://www.ibm.com/servers/eserver/clusters/>

### 1.2.3 The new IBM eServer Cluster 1350

The IBM @server Cluster 1350 is a new Linux cluster offering. It is a consolidation and a follow-on of the IBM @server Cluster 1300 and the IBM xSeries “custom-order” Linux cluster offering delivered by IGS. This new offering provides greater flexibility, improved price/performance with Intel Xeon™ processor-based servers (new xServer models x345 and x335), and the superior manageability, worldwide service and support, and demonstrated clustering expertise that has already established IBM as a leader in Linux cluster solutions.

The Cluster 1350 is targeted at the High-Performance Computing market, with its main focus on the following industries:

- ▶ Industrial sector: Petroleum, automotive, aerospace
- ▶ Public sector: Higher education, government, research labs
- ▶ Life sciences
- ▶ Financial services
- ▶ Service providers
- ▶ Communications/media: For Web server farms, e-business infrastructures, collaboration, and digital content creation

Also, with its high degree of scalability and centralized manageability, the Cluster 1350 is ideally suited for Grid solutions implementations.

For more information about the new IBM @server Cluster 1350, refer to the following Web site:

<http://www.ibm.com/servers/eserver/clusters/>

For more information about xServer models x345 and x335, go to the following Web site:

<http://www.pc.ibm.com/us/eserver/xseries/>

**Restriction:** At the time this book was written xCAT tool did not support the new Intel Xeon processor-based servers (xServer models x345 and x335) offered by the new Cluster 1350.

For more information about xCAT go to:

<http://x-cat.org/>

## 1.3 Making up an HPC cluster

An High-Performance Computing cluster typically has a large number of computers (often called nodes) and, in general, most of these nodes would be configured identically. The idea is that the individual tasks that make up a parallel application should run equally well on whatever node they are dispatched on.

However, some nodes in a cluster often have some physical and logical differences. In the following sub-sections we discuss logical node functions and then physical node types.

### 1.3.1 Logical functions that a node can provide

As we stated before, a cluster is two or more (often many more) computers working as a single logical system to provide services. Though from the outside the cluster may look like a single system, the internal workings to make this happen can be quite complex.

Figure 1-3 on page 8 presents the logical functions that a physical node in a cluster can provide. Remember, these are logical functions; in some cases, multiple logical functions may reside on the same physical node, and in other cases, a logical function may be spread across multiple physical nodes.

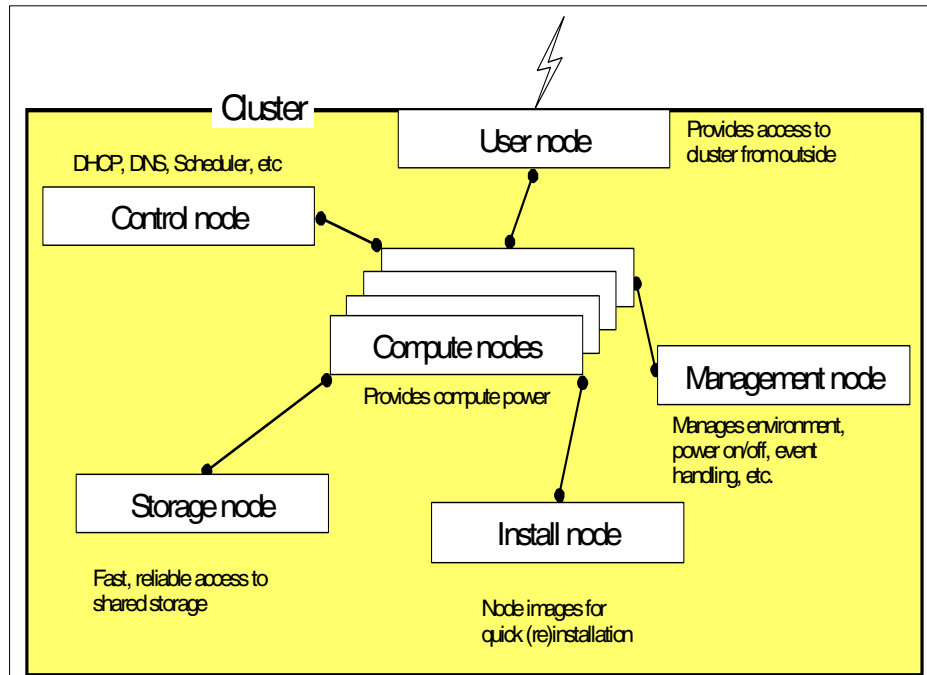


Figure 1-3 Logical structure of a cluster

## Compute node

The compute node is where the real computing is performed. The majority of the nodes in a cluster are typically compute nodes. In order to provide an overall solution, a compute node can execute one or more tasks, based on the scheduling system.

## Management node

Clusters are complex environments, and the management of the individual components is very important. The management node provides many capabilities, including:

- ▶ Monitoring the status of individual nodes
- ▶ Issuing management commands to individual nodes to correct problems or to provide commands to perform management functions, such as power on/off

You should not underestimate the importance of cluster management. It is an imperative when trying to coordinate the activities of a large numbers of systems.



## **Install node**

In most clusters, the compute nodes (and other nodes) may need to be reconfigured and/or reinstalled with a new image relatively often. The install node provides the images and the mechanism for easily and quickly installing or reinstalling software on the cluster nodes.

## **User node**

Individual nodes of a cluster are often on a private network that cannot be accessed directly from the outside or corporate network. Even if they are accessible, most cluster nodes would not necessarily be configured to provide an optimal user interface. The user node is the one type of node that is configured to provide that interface for users (possibly on outside networks) who may gain access to the cluster to request that a job be run, or to access the results of a previously run job.

## **Control node**

Control nodes provide services that help the other nodes in the cluster work together to obtain the desired result. Control nodes can provide two sets of functions:

- ▶ Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and other similar functions for the cluster. These functions enable the nodes to easily be added to the cluster and to ensure they can communicate with the other nodes.
- ▶ Scheduling what tasks are to be done by what compute nodes. For instance, if a compute node finishes one task and is available to do additional work, the control node may assign that node the next task requiring work.

## **Storage node**

For some applications that are run in a cluster, compute nodes must have fast, reliable, and simultaneous access to the storage system. This can be accomplished in a variety of ways depending on the specific requirements of the application. Storage devices may be directly attached to the nodes or connected only to a centralized node that is responsible for hosting the storage requests.

## 1.3.2 xSeries models used in our cluster

For the purpose of this book we used the IBM eServer xSeries Model 342 and Model 330 physical nodes that make up a Cluster 1300. However, other types of models can be available (such as Model 335 and Model 345), so for more information about the xSeries models suitable for building a cluster contact your local IBM representative and also refer to the xSeries Intel processor-based servers Web page at:

<http://www.pc.ibm.com/us/eserver/xseries/>

For xCAT standpoint, all xSeries and Intellistations are supported. The support of other compute and non-compute hardware (for example, Fibre controllers, etc.) can be done using xCAT's APC MasterSwitch and MasterSwitch+, Baytech, and Intel EMP methods. As many Linux clusters are the extension of existing clusters, one of the goals of xCAT was to manage those clusters.

In a typical IBM Linux cluster configuration, shown in Table 1-1, we have three types of nodes, management nodes, compute nodes, and storage nodes. Note that more than one function can be provided by one single node. The final cluster architecture must consider the application the customer wants to run and the whole solution environment.

Table 1-1 Typical Linux cluster

Generic node type	Functions	xSeries models
Management (aka, master)	Management Install Control User	Model 342 Model 345
Compute	Compute	Model 330 Model 335
Storage	Storage	Model 342 Model 345

### Management node (master node)

The term *management node* is a generic term. It is also known as *master node*. This node aids in controlling the cluster but can also be used in additional ways.

Management nodes generally provide one or more of the following logical node functions described in the last section:

- ▶ Management node
- ▶ Installation node
- ▶ User node
- ▶ Control node

In a small cluster, say eight compute nodes, all of these functions can be combined in one management node. In larger clusters, the functions are probably split across multiple machines for security and performance reasons.

### ***Model 342***

The Model 342, shown in Figure 1-4 and used as the management node, is a 3U rack-optimized server with one or two Intel Pentium III processors and 5 full-size PCI slots. The node also includes an optional Remote Supervisor Adapter (RSA) card for hardware control and management.



*Figure 1-4 Model 342 management node*

## **Compute nodes**

The compute nodes form the heart of the cluster. The user, control, management, and storage nodes are all designed to support the compute nodes. It is on the compute nodes that most computations are actually performed. These nodes are logically grouped, depending on the needs of the job and as defined by the job scheduler.

### ***Model 330***

The Model 330, shown in Figure 1-5 and used as a compute node, is a 1U rack-optimized server with one or two Intel Pentium III processors and two PCI slots (one full-length and one half-length). One in every eight Model 330 nodes must have the Remote Supervisor Adapter included to support the cluster management network.



Figure 1-5 Model 330 for compute nodes

## Storage nodes

Often when discussing cluster structures, a storage node (typically a Model 342 or a Model 345) is defined as a third type of node. However, in practice a storage node is often just a specialized version of a node. The reason that storage nodes are sometimes designated as a unique node type is that the hardware and software requirements to support storage devices might vary from other management or compute nodes. Depending on your storage requirements and the type of storage access you require, this may include special adapters and drivers to support the attached storage devices.

### 1.3.3 Other cluster components

Aside from the cluster nodes (management node, compute nodes, and storage nodes) that make up a cluster, there are several other key components that must also be considered. The following sub-sections discuss some of these components.

#### Ethernet switch

10/100 Ethernet switches are included to provide the necessary node-to-node communication. Basically, we need two types of LANs (or VLANs); one for management and another for application. They are called management VLAN and cluster VLAN, respectively. One Ethernet switch per rack is required. For more information about VLANs see Table 2-3 on page 25.

#### Myrinet switch

Some clusters need high-speed network connections to allow cluster nodes to talk to each other as quickly as possible. The Myrinet network switch and adapters are designed specifically for this kind of high-speed and low-latency requirement.

More information about Myrinet can be found at the Myricom Web site at:

<http://www.myri.com/>

## Terminal server

Terminal servers provide the capability to access each node in the cluster as if using a locally attached serial display. The BIOS of compute and storage nodes in xSeries clusters are capable of redirecting the machine POST out of the serial port. After POST the boot loader and operating system also utilize the serial port for display and key input. The terminal servers provide cluster operators the ability to use common tools such as **telnet**, **rsh**, or **ssh** to access and communicate with each node in the cluster and, if necessary, multiple nodes simultaneously. Cluster management packages then have the ability to log whatever the nodes redirect out of the serial port to file, even while not being viewed by the operator. This gives the operator an out-of-band method of viewing and interacting with the entire boot process of a node from POST to the operating system load. This is useful for debugging the boot process, performing other low-level diagnostics, and normal out-of-band console access. Terminal servers provide this capability by allowing a single display to be virtually attached to all nodes in the cluster. Terminal servers from Equinox and iTouch Communications are examples of such devices and are commonly used in clusters.

More information on terminal servers from Equinox can be found at the following Web site:

<http://www.equinox.com/>

More information on terminal servers from iTouch can be found at the following Web site:

[http://www.mrv.com/products/remote\\_management/products](http://www.mrv.com/products/remote_management/products)

## Keyboard, video, mouse switch

In addition to the terminal servers (for serially attached displays and printers), it is also not practical to provide a keyboard and a mouse for every node in a cluster. Therefore, using a common keyboard, video, mouse (KVM) switch is also indispensable when building a cluster. The Model 330 includes a Cable Chain Technology (C2T) interface that allows a KVM chaining in one single cable, as presented in Figure 1-6. This allows an operator to attach to any individual node and perform operations if required.

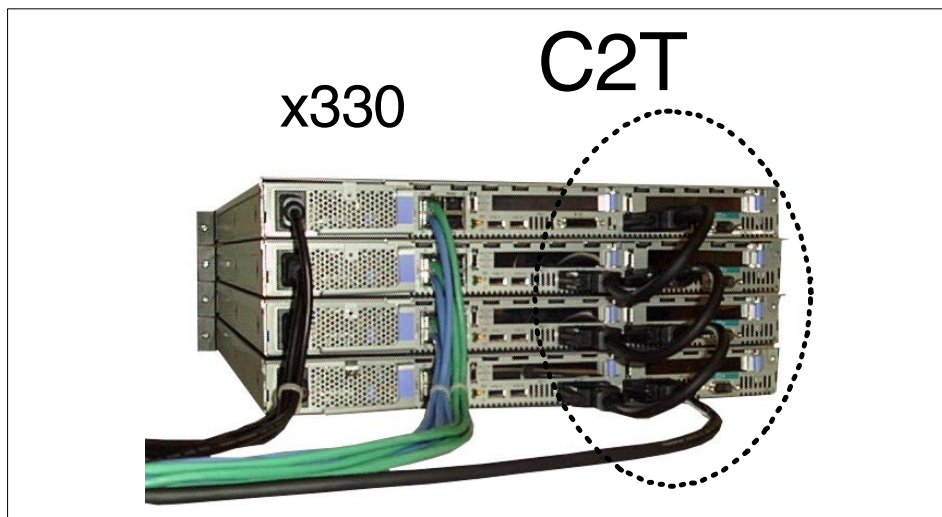


Figure 1-6 Cable chain technology

### Management Processor Network

Each Model 330 node has a management processor (also known as a service processor) that allows remote node power on/off/reset capability; monitors node environmental conditions (such as: fan speed, temperature, and power); and allows remote POST/BIOS console, power management, and SNMP alerts. The Model 342 nodes must include an optional Management Processor Adapter in order to provide the same functionality. Access to the node service processors is provided through an RS485 daisy-chain network. As seen in Figure 1-7, up to eight nodes per chain is the technical recommendation. The first node in the daisy-chain is then connected via the Ethernet through a Remote Supervisor Adapter (RSA). This forms a management processor daisy-chain network. The RSA is a half-length PCI adapter, is externally powered, and occupies a PCI slot in the first node in the chain. The Management Processor Adapter used on x342 nodes is the RSA card.

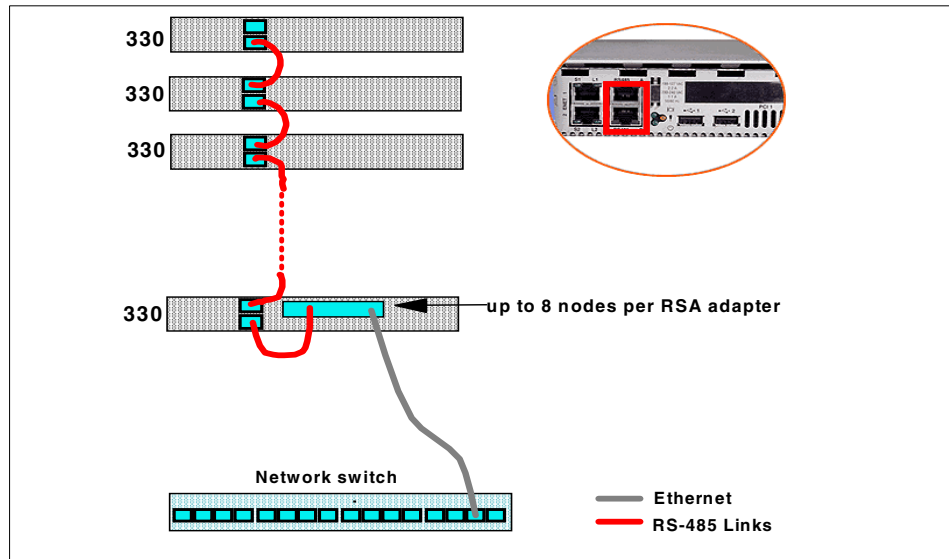


Figure 1-7 Management processor network

## 1.4 Software

A general cluster system management tool provides facilities to build, manage, and expand clusters efficiently. Cluster-ready software from IBM enables a cluster to look and act like a single system for end users and system administrators.

Although you can build your cluster by using different cluster system management tools, such as the IBM Cluster Systems Management (CSM) and the Open Source Cluster Application Resources (OSCAR), this book is primarily focused on the xCAT cluster management toolkit for Red Hat distribution.

### 1.4.1 IBM Cluster Systems Management for Linux

IBM Cluster Systems Management for Linux (CSM) provides a distributed system-management solution for machines, or nodes, that are running the Linux operating system. CSM is an integral part of the IBM @server Cluster 1300 platform for deploying Linux applications requiring a cluster, and is also available as a separately orderable software product. With this software, an administrator can easily set up and maintain a

Linux cluster by using functions like automated set-up, hardware control, monitoring, and configuration file management. The concepts and software are derived from IBM Parallel System Support Programs for AIX (PSSP), and from applications available as open source tools.

These are some CSM capabilities:

- ▶ Install Linux and CSM client on cluster nodes over the network
- ▶ Add, remove, or change nodes
- ▶ List nodes (with persistent configuration information displayed about each node in the list)
- ▶ Run remote commands across nodes or node groups in the cluster
- ▶ Gather responses from the above commands
- ▶ Monitor nodes and applications as to whether they are active
- ▶ Monitor CPU, memory, and system utilization
- ▶ Run automated responses when events occur in the cluster
- ▶ File manager configurator to enable an administrator to set up cluster-wide files in a central location

For more information about CSM, please refer to *Linux Clustering with CSM and GPFS*, SG24-6601, and to the following Web site:

<http://www.ibm.com/servers/eserver/clusters/library/>





## xCAT introduction

This chapter provides a short introduction to xCAT. We discuss:

- ▶ What xCAT is
- ▶ Downloading xCAT
- ▶ Installing a Linux cluster with xCAT
- ▶ Planning
- ▶ Preparing hardware
- ▶ Installing a management node
- ▶ Installing cluster components and compute nodes

**Attention:** Extreme Cluster Administration Toolkit (xCAT) is a set of shell scripts that automate some processes. The scripts are provided *as-is*. IBM has no obligation to provide any error correction or enhancements. Further, IBM makes no representations or warranties, express or implied, including the implied warranties of merchantability and fitness for a particular purpose with respect to the scripts or their use, nor shall IBM have any liability in respect to any infringement of any intellectual property rights of third parties due to customer's operation under the licenses or rights herein granted.

**Attention:** Linux and other open source programs (OSPs), when packaged with or preloaded by IBM on any computer system, are distributed and licensed to you by Caldera Inc., Red Hat Inc., SuSE GMBH, TurboLinux Inc., or other distributors of OSP, not IBM.

No express or implied patent, copyright, or other license is granted to you by IBM for the use of Linux or other OSPs.

IBM delivers Linux and other OSPs *as-is*, without any warranty.

IBM is not responsible for any claim that Linux or other OSP infringe a third party's patent or copyright.

## 2.1 What xCAT is

Setting up the installation and management of a cluster is a complicated task, and doing everything by hand can become very complicated. The development of xCAT grew out of the desire to automate a lot of the repetitive steps involved in installing and configuring a Linux cluster.

The development of xCAT is driven by customer requirements and because xCAT itself is written entirely using scripting languages such as korn shell, perl, and Expect. The administrator can easily modify the scripts should the need arise. All third party products used in xCAT are freely available; some of them are OSS, and others are available from IBM at no charge.

- ▶ Automated installation
  - Network booting with PXE or Etherboot/GRUB
  - Red Hat installation with Kickstart
  - Other OS installation using imaging or cloning
  - Nodes automatically configured
  - Errata automatically installed
- ▶ Hardware management and monitoring
  - Supports the Advanced Systems Management features in IBM @server xSeries
    - Remote power control
    - Remote vital statistics (fan speed, temperatures, voltages)
    - Remote inventory (serial numbers, BIOS levels)
    - Hardware event logs
    - Hardware alerts via SNMP
  - Supports remote power control switches for control of other devices
    - APC MasterSwitch
    - BayTech Switches
    - Intel EMP
  - Traps SNMP alerts and notify administrators via e-mail
- ▶ Software administration
  - Parallel shell to run commands simultaneously on all nodes or any subset of nodes
  - Parallel copy and file synchronization
  - Assists with the installation and configuration of the HPC software stack
    - MPICH and LAM for parallel applications using message passing
    - Maui and PBS for scheduling and queuing of jobs
    - GM for fast and low latency inter-process communication using Myrinet

- ▶ Remote console support for text and graphics
  - Terminal servers for remote console
    - Equinox ELS and ESP
    - iTouch In-Reach and LX series
    - Remote Console Redirect feature in IBM @server xSeries BIOS
  - VNC for remote graphics

### 2.1.1 Download xCAT

xCAT is available at no charge to IBM Linux cluster customers. To download xCAT, you need a user ID and password. To apply for a user ID, visit <http://x-cat.org/download/>. If you have a user ID, you can download the latest release of xCAT from:

<http://x-cat.org/download/xcat/>

### 2.1.2 Directory structure

This is an overview of the directories in the xCAT distribution:

<b>bin/</b>	Commands for all architectures.
<b>build/</b>	Patches and scripts to build third-party packages; scripts to automate installation and patching of common cluster software packages such as PBS, MAUI, MPICH. It also contains scripts for installation of open source packages used to manage the cluster, such as Etherboot, Conserver, and ATFTP.
<b>clone/</b>	Cloning support. Boot images and startup scripts used to clone nodes disks with xCAT. This is primarily used for operating systems where the partitions/file systems cannot be imaged.
<b>doc/</b>	Documentation. Text, PDF files, and HTML documentation of xCAT installation and advanced features.
<b>elilo/</b>	elilo support (the boot loader for Itanium-based operating system). It contains elilo code and configuration scripts
<b>etc/</b>	Primary location for configuration files and tables used by xCAT commands.
<b>Etherboot/</b>	Contains CD and floppy Etherboot images for IBM nodes using AMB and Intel NICs that do not support PXE.

<b>flash/</b>	Contains scripts, and BIOS update and ServeRAID update images used with PXE's memdisk option to enable updates and configuration nodes remotely and in parallel.
<b>i686/</b>	Binary commands and libraries specific for i686 architecture. Contains precompiled libraries and binaries for open source packages used to manage the cluster for the Intel Pentium III processor and the Intel Pentium 4 processors.
<b>ia64/</b>	Contains precompiled libraries and binaries for open source packages used to manage the cluster for the Intel Itanium processor.
<b>image/</b>	Boot images and startup scripts used to image nodes disk partitions with xCAT. This is primarily used for operating systems where the partition/file system types can be managed; used primarily for Windows NTFS nodes.
<b>ksXX/</b>	Contains template Kickstart configuration files and PXE boot information for specific Red Hat distributions.
<b>lib/</b>	Contains low-level scripts (libraries for all architectures) called by commands in bin/.
<b>man/</b>	Contains man page data for many xCAT commands and tables.
<b>pbs/</b>	Supports files for PBS. Epilogue and prologue scripts used by PBS when configured by xCAT scripts.
<b>post/</b>	Files for post-installation modifications and updates of the installed nodes.
<b>rc.d/</b>	Contains default System V scripts for many services utilized by management and cluster packages contained in xCAT.
<b>samples/</b>	Sample files for third party applications. Configuration files for management and cluster packages used by xCAT as well as sample /etc xCAT tables.
<b>sbin/</b>	Commands for setting up the cluster (node) installation environment.
<b>src/</b>	Source code for third-party applications. Cluster software and open source management packages supported by xCAT.
<b>stage/</b>	Stripped staging image for node installation and scripts to generate the complete staging images.

<b>tftp/</b>	Contains elilo, pxelinux, and pxelinux default configuration files, used during network booting.
<b>windows/</b>	Contains scripts used when cloning/imaging Windows nodes to generate unique SIDs and set host-specific information.

## 2.2 Installing a Linux cluster with xCAT

If you are installing a large number of computers, whether they are going to be part of a cluster or not, you do not want to have to install and configure every computer by hand. You want to reduce the number of manual steps required as much as possible and control the whole process from the management node.

A typical cluster installation will go through the following steps:

1. Planning
2. Hardware preparation
3. Management node installation
4. Cluster installation

### 2.2.1 Planning

In the following sections we discuss planning.

#### Naming convention

Every node and device in the cluster needs a name. xCAT supports any kind of naming convention, and we recommend that you set up one to help to avoid confusion. Once you decide on a naming convention, use it consistently throughout the cluster.

The naming convention we use in our example is simply the name of the node or device, followed by a sequential number padded to three digits, optionally followed by a hyphen and the name of an interface.

*Table 2-1 Naming convention*

Node or device name	Description
masternode	The name of the management node
masternode-eth0	First Ethernet interface on the management node
masternode-eth1	Second Ethernet interface on the management node

Node or device name	Description
node001 node002	Compute nodes
storage001	Storage node
node001-myri1 node002-myri1	Myrinet interface on a compute node
rsa001 rsa002	Remote Supervisor Adapters
els001 els002	Equinox ELS terminal servers
cisco001	Cisco 3525XL Ethernet switch
myri001	Myrinet switch
apc001	APC MasterSwitch

## IP address assignments

Most clusters will use private IP addresses on the cluster, and management and IPC VLANs. This is not always the case, depending on the customer requirements. As shown in Figure 2-1, we used part of the IP address to identify the type of the VLAN and the rack number where the node is located.

10	VLAN Type	Rack Number	Node Address
0 = Cluster VLAN 1 = Management VLAN 2 = IPC			

Figure 2-1 IP address octets

At this point, you will also need to decide which Ethernet interface is connected to which VLAN and make sure that the cluster is cabled accordingly.

Table 2-2 IP address assignments

Node or device name	IP address	Comments
<b>Cluster VLAN - Subnet 10.0.0.0/16</b>		
masternode	10.0.0.1	Gigabit interface in slot 3 on management node.

Node or device name	IP address	Comments
node001	10.0.1.1	The third octet in the IP address indicates the rack where the node is located.
node002	10.0.1.2	
node003	10.0.1.3	
storage001	10.0.1.9	Storage node.
<b>Management VLAN - Subnet 10.1.0.0/16</b>		
masternode-eth1	10.1.0.1	Ethernet interface in slot 2 on management node.
rsa001	10.1.1.101	The third octet in the IP address indicates the rack where the equipment is located.
rsa002	10.1.1.102	
els001	10.1.1.161	
apc001	10.1.1.181	
myri001	10.1.1.201	
cisco001	10.1.1.241	
<b>IPC VLAN - Subnet 10.2.0.0/16</b>		
node001-myri0	10.2.1.1	The third octet in the IP address indicates the rack where the node is located.
node002-myri0	10.2.1.3	
node003-myri0	10.2.1.3	
<b>Public VLAN - Subnet 192.168.42.0</b>		
masternode-eth0	192.168.42.1	Onboard Ethernet interface on management node.

## VLAN assignments

A cluster typically has an internal network and a connection to an external network. The external network is often referred to as the public VLAN. The internal network is normally partitioned into VLANs.

- ▶ The *public VLAN* allows users to access the cluster via the management node or, optionally, the user nodes. The cluster administrators can also use the public VLAN for remote management.



- ▶ The *cluster VLAN* provides the internal network for all of the nodes in the cluster. All of the nodes in the cluster will use the cluster VLAN for network booting and network installation, so it is important that you use a network adapter that is capable of network booting via PXE to connect to this VLAN. On a working cluster, the cluster VLAN is also used for in-band management, for normal network traffic between the nodes, and to start jobs on the nodes, etc. The cluster VLAN is typically a 100 Mbit/s Ethernet network. The management node, and any installation nodes and storage nodes are usually given a 1000 Mbit/sec (gigabit) network connection to the cluster VLAN.
- ▶ The *management VLAN* connects all the management devices to the management node. This includes the MPA cards, the APC switches, the terminal servers, and the management interfaces of the Cisco and Myrinet switches. It is possible to combine this with the cluster VLAN if you want, but we recommend that you always use, for security reasons, a separate and isolated management VLAN. The management VLAN is used for out-of-band management, so even when there is a network problem on the cluster VLAN, you may be able to use the management VLAN to find and correct the problem. The management VLAN is typically a 10/100Mbits Ethernet network.
- ▶ The *Inter-Process Communication VLAN* or IPC VLAN is an optional network used by the cluster applications for message passing with MPI or to access storage on a storage node. For this reason, the cluster VLAN will normally be a high-speed, low-latency network like Myrinet, so that the network does not become a bottleneck for parallel programs or accessing storage. If this is not required, you would normally use the cluster VLAN to do IPC.

Before you continue, you should answer the following questions.

- ▶ Are you going to partition your cluster network in VLANs?
- ▶ How many VLANs and which VLANs are you going to use?
- ▶ Which ports on the switch will be assigned to which VLAN?
- ▶ Which IP subnet will be used on each VLAN?

Table 2-3 VLAN assignments

VLAN	IP subnet	VLAN number	Switch ports
Cluster	10.0.0.0/16	VLAN002	FastEthernet 1–9 Gigabit Ethernet 2
Management	10.1.0.0/16	Default	FastEthernet 10–20
Public	192.168.42.0	VLAN003	FastEthernet 21–24

**Note:** The IPC VLAN is a completely separate Myrinet network and, as such, not strictly speaking, a VLAN.

Also, in some cluster installations, the public interface of a management or user node is connected directly to the public network, so there is no need to configure a public VLAN on the switches in the cluster, but it is still referred to as the public VLAN.

## 2.2.2 Hardware preparation

Building a cluster can be a very long and demanding process. If you order all the components individually, it will take a lot of organization and effort to bring them all together and create a cluster. Not to mention the mountain of cardboard that is left after unpacking all the parts.

To avoid all these problems, you can use integration services from IBM. IBM will help you create the hardware configuration and will build the cluster at an IBM location before shipping it completely cabled, directly to your computer room. IBM will also test the cluster as a complete system, and not just the individual components.

See Chapter 3, “Hardware preparation” on page 31, for detailed information.

## 2.2.3 Management node installation

Once you have your cluster all set up in the computer room, you are ready to install the management node. The management node will also be an installation server, which allows you to install all the other nodes in the cluster automatically.

Installation of the management node typically involves the following steps:

1. Install the Red Hat Linux operating system.
2. Install the errata for Red Hat Linux.
3. Configure networking.
4. Install and configure xCAT.
5. Populate the xCAT tables.
6. Configure management node services.
  - System logging
  - DNS for name resolution
  - DHCP for dynamic network configuration
  - TFTP to support network booting
  - NFS to support network installation
  - NTP to keep time synchronized in the cluster

- SSH for remote control of the nodes
- SNMP for hardware alerts

See Chapter 4, “Management node installation” on page 43, and Chapter 5, “Management node configuration” on page 57.

## 2.2.4 Cluster installation

Once the management node is fully configured, we need to configure and install the other components in the cluster. This is explained in detail in Chapter 6, “Cluster installation” on page 83.

The installation process for the cluster happens in four stages.

1. Stage 1: Hardware setup
2. Stage 2: MAC address collection
3. Stage 3: Management processor setup
4. Stage 4: Node installation

### Stage 1: Hardware setup

In this stage we set up all the hardware components in the cluster. The hardware setup can be done before or during the management node configuration, but for most of the devices, xCAT provides help in the form of a setup script. Because this removes the need for manual procedures to configure the hardware, it is convenient to start the hardware setup when the management node is available.

### Stage 2: MAC address collection

When building a cluster using a large number of identical systems for the compute nodes, it is important for management that you know the physical location of every node in the cluster. For this reason, we need a process that allows you to assign node names to every node based on their physical location. Because each node has a different MAC address for the primary Ethernet adapter, we can use this to uniquely identify every node. During MAC address collection, we automatically determine the physical location for each node and save the relationship between MAC address and physical location in a table. The DHCP server is then configured to assign the correct IP address and host name to every node, based on the MAC address used in the DHCP request.

We determine the physical location of each node based on the port of a terminal server or Ethernet switch that was used to connect the serial port or the Ethernet port of that node. For example, node 1 is connected to port 1 on the Cisco switch or port 1 of the Equinix terminal server.

### Stage 3: Management processor configuration

This stage is only necessary for clusters that have nodes using the IBM Advanced Systems Management features. During stage 3, the management processors in all the nodes are configured.

- ▶ The management processor is given a name and a numeric ID so that it can be addressed over the management processor network.
- ▶ Alerts are enabled so that hardware failures are reported via the management processor network.
- ▶ The management serial port is configured to the correct speed settings so that you can access it via the terminal server.
- ▶ The default user ID and password is set.

### Stage 4: Node installation

The nodes in the cluster are installed and configured automatically with a single command from the management node. During installation, all the nodes boot from the network and start the installation automatically. The installation process uses a standard Red Hat Kickstart installation over NFS. Figure 2-2 on page 30 illustrates the process booting a node from the network, running the install over NFS and configuring the node to boot from the local hard drive after the installation.

In a cluster, all the nodes are always configured to boot from the network first. This allows the administrator to control the boot process of each node from the management node. Each node has a network loader configuration file, which is manipulated by xCAT to control what the node is going to boot during the next reboot. The configuration files are stored in `/tftpboot/pxelinux.cfg/`. In case of a network boot problem (PXE boot fail), the hard drive boots by default.

When a node boots from the network, it does a DHCP request to get the network configuration. The DHCP server returns the IP address and other network parameters for the node, but it also includes a bootfile parameter and the IP address of the TFTP server where the bootfile can be loaded from. The bootfile used to boot Linux is called `pxelinux.0`.

The node downloads `pxelinux.0` via TFTP and starts it. `pxelinux.0` will then start to look for a configuration file to download. It does this by converting the IP address to an eight-digit hexadecimal number. It then repeatedly tries to download this file from the TFTP server, removing the last digit every time the file is not found. If PXELinux cannot find any files, it tries to load a file called *default*. Example 2-1 on page 29 shows the output from `pxelinux.0` booting with the IP address 10.9.0.1.

*Example 2-1 pxelinux configuration*

---

```
Trying to load: pxelinux.cfg/OA090101  
Trying to load: pxelinux.cfg/OA09010  
Trying to load: pxelinux.cfg/OA0901  
Trying to load: pxelinux.cfg/OA090  
Trying to load: pxelinux.cfg/OA09  
Trying to load: pxelinux.cfg/OA0  
Trying to load: pxelinux.cfg/OA  
Trying to load: pxelinux.cfg/0  
Trying to load: pxelinux.cfg/default
```

---

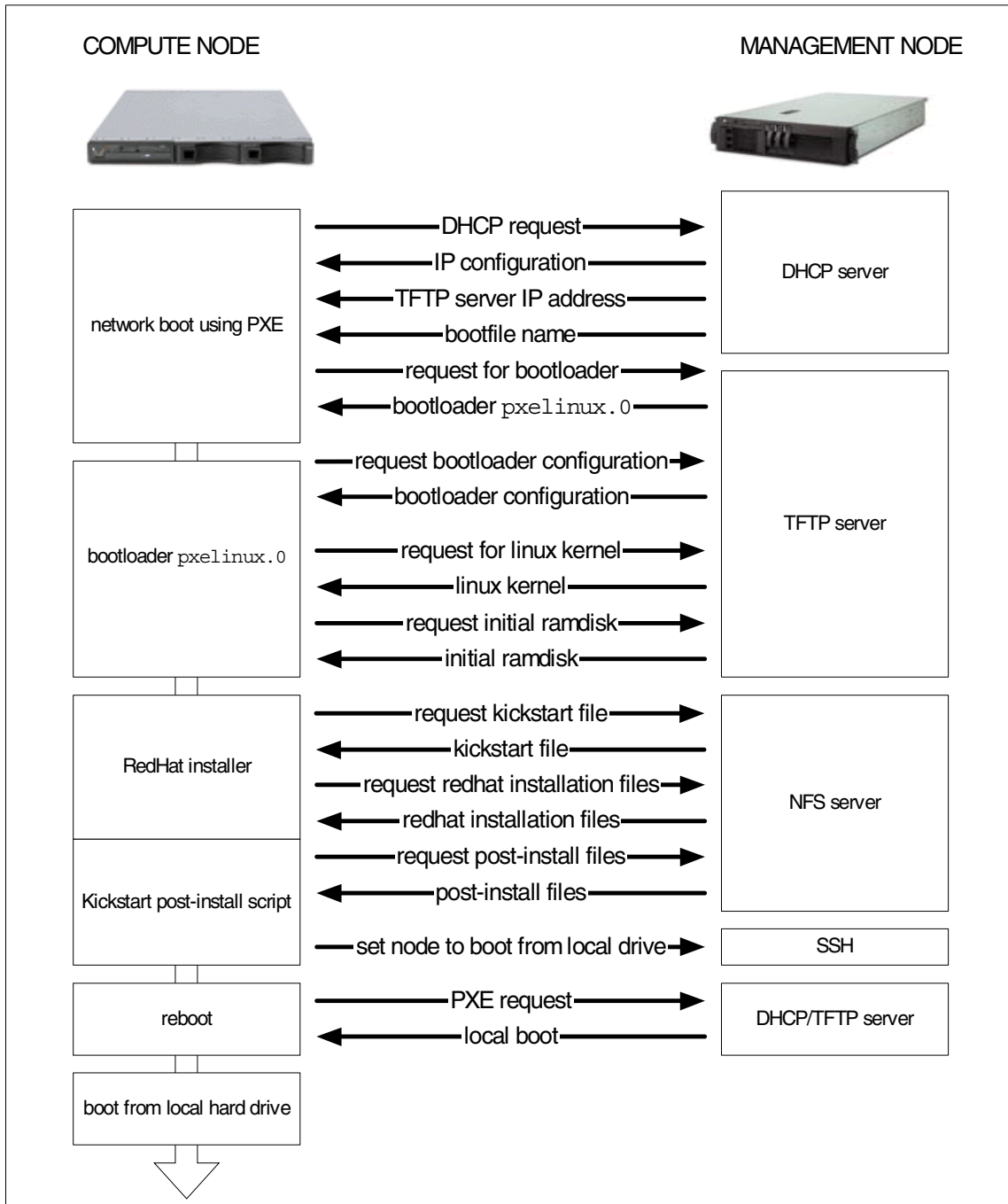


Figure 2-2 Network boot and installation process



## Hardware preparation

This chapter describes the basic hardware setup, configuration, and customization. We discuss in detail how to install the equipment, add additional components such as the Myrinet cards, and populate and cable the rack(s). After completing this, you can start installing the software to get your cluster up and running. It is a good idea at this stage to have all of your planning details for your cluster at hand, so that you can implement the steps as you work through the chapter.

We give detailed examples for the eight compute node cluster we used in our lab. As discussed in the previous chapters, there are multiple options on how to design and implement a cluster. You might want to use only one large Myrinet switch and one large FastEthernet or Gigabit Ethernet switch for the whole cluster, or use individual interconnected switches for each rack. Whenever we talk about switches you need to remember that there might only be one switch in your setup.

The topics covered in this chapter include:

- ▶ Preparing the node hardware
- ▶ Populating the rack(s)
- ▶ Connecting the cables

## 3.1 Node hardware installation

The first thing to do is to prepare each individual node before it goes into the rack. This means installing the usual additional options, such as a second CPU, memory, and hard disks. If your cluster was purchased with integration services, this work has already been done for you, but you should understand the steps involved in case hardware components need to be added or replaced.

### Management node (x342)

We use the built-in Ethernet port for connecting to the customer network, so we add another FastEthernet card for the management processor network (MPN) link. For better availability, a ServeRAID controller is installed. Finally, we need a Gigabit Ethernet adapter for the uplink to the FastEthernet switch to achieve fast compute node network installation and cluster scalability.

For performance reasons you may want to place the I/O intensive adapters (mainly ServeRAID and Gigabit Ethernet) into specific slots so that they are on different peripheral computer interface (PCI) busses. For details please refer to your system documentation or the labels on the back of your server.

### Compute nodes (x330)

If Myrinet or Gigabit Ethernet is being used for the IPC network, you will install a network interface card (NIC) into the long PCI slot (slot 1). Finally, you need to install a Remote Supervisor Adapter card (RSA) card into the remaining short PCI slot (slot 2) on every eighth node. An x330 with Myrinet and RSA cards installed is shown in Figure 3-1 on page 33.

**Note:** Older versions of the x330 BIOS firmware required you to move a serial port to be moved from the default COM A port to COM B. With current BIOS levels (1.06 for the 8654, 1.03 for the 8674), this is no longer necessary or desirable, as it requires opening every compute node to move the jumper and may interfere with the use of console redirection. All nodes must use the same serial port for their console port, as this is defined globally in site.tab.





Figure 3-1 x330 with PCI cards installed

## 3.2 Populating the rack and cabling

Before placing the nodes into the racks, you should know the position of your additional equipment, such as network switches, KVM switches, terminal servers, and so on. It would be very helpful to have a plan for the rack that was derived from the rack configurator or sketched out by hand during your planning phase.

When placing equipment in the rack, you need to consider cable organization, weight, and heat distribution. Heavy hardware, such as the management node, storage arrays, and the uninterruptable power supply (UPS), should be placed in the bottom. Consider the direction of air flow through each unit. Equipment that must exhaust out the front of the rack is best placed at the top to avoid situations that lead to heated air from one unit being drawn into the intake of another. Empty slots in the rack should also be covered with blanking panels to prevent heated air from being drawn forward. The power distribution units (PDUs) are

installed on the right-hand side of the racks (seen from the front of the rack), as most power supplies are on this side, too. This leaves the left-hand side for the KVM switch or front end PDUs when three or more Netfinity back-end PDUs are used.

**Tip:** Be sure to record the hardware MAC addresses for devices that have one recorded on the case before mounting the units into the rack. Depending on the device, these may be used for DHCP configuration.

Additionally, it is important to leave enough space for proper air flow and maintenance work.

Next, put your nodes into the rack and label them clearly.

- ▶ One node ID label on the front of the machine
- ▶ One node ID label on the rear, preferably on the left-hand side (for x330 you will need to label the mounting rail as there is not enough space on the rear of the node)

**Attention:** Some customers in secure facilities have pointed that you should label the rack, not the nodes, because the nodes get replaced. For customers that require a label on the node for an asset tag, avoid labeling the CDROM or the hard driver since they are removable (and can be broken off).

Once you have finished labeling, you can start to cable the rack. This includes labeling each cable that is longer than 1 foot or 30 centimeters at each end and using different colors for different kinds of cables. For labeling the cables we recommend the use of a professional wire marker printer, such as the Brady I.D. Pro Plus Wire Marker Printer, or any other.

In the following pictures, the cable management of the x342 and the Netfinity racks (with rack extensions) is not demonstrated. Nevertheless, we highly recommend that you use their cable management features to achieve less confusion regarding your cluster's cabling.

1. Connect the power cables (including the external RSA AC adapter and additional devices such as switches) to the PDUs or APC MasterSwitches (whichever your rack is equipped with). If you are using an APC MasterSwitch, it is important to make notes of which device is plugged into each numbered port on the switch; you will need this information later to configure xCAT's power control tables. Also, you need to collect the MAC address so you can use DHCP. Note that the MAC address is not on the unit, but on a piece of paper with the APC manual.

2. Connect your rack-mounted monitor, mouse, and keyboard to the KVM switch of the rack.
3. Connect the C2T cables from node to node, daisy chaining them, and connect the C2T keyboard, video, mouse splitting cable to the last C2T OUT port, which is then connected to the KVM switch of your rack using the KVM (NetBAY console) cable.
4. Connect the RS485 cables in daisy chains of eight nodes, plugging the last cable of each chain into the RSA card dongle, which is directly connected to the RSA card of that group. Use the single-tailed dongle for x330 nodes.

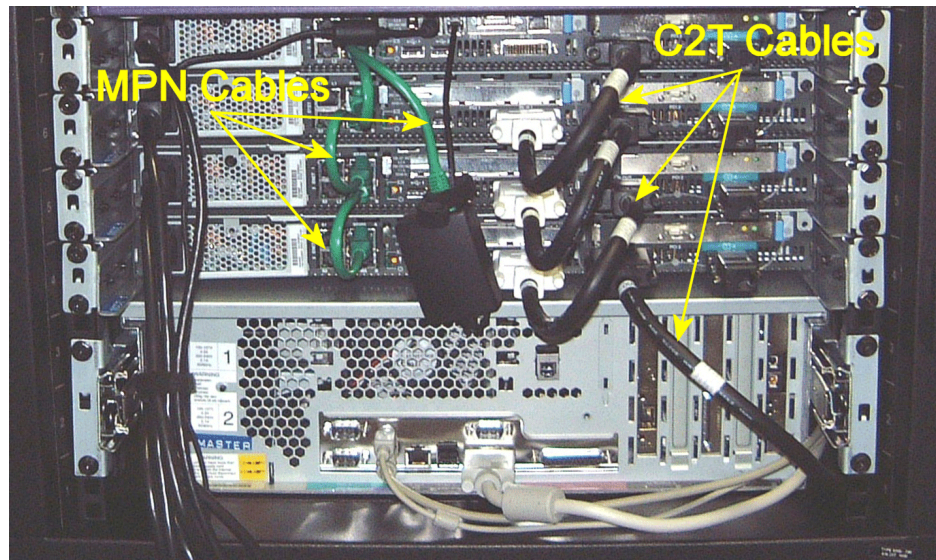


Figure 3-2 MPN and C2T cabling

Your rack should now look somewhat like Figure 3-2. After powering on the nodes (especially the ones with the RSA cards inside), you can switch to their screens via C2T and the KVM switches.

1. Connect the terminal servers to the Ethernet switch.
2. Connect the MPN card's Ethernet interfaces to the Ethernet switch.



3. Connect the nodes to the terminal servers (ELS) using the serial-to-RJ45 converters and RJ-45 cables (see the left-hand side of Figure 3-3 on page 36). Note that while these serial cables use the same RJ-45 connectors as Ethernet, the cables are not interchangeable. It is important to make note of which port each node is connected to, and it is highly recommended that you cable the nodes into sequential ports (node 1 into port 1, node 2 into port 2, and so forth), and save the higher numbered ports for the management equipment.
4. Connect the Ethernet cables from the nodes to the switches and the Ethernet cable from the Myrinet switch management port to the Ethernet switch (see the right side of Figure 3-3). Here it is also important to maintain a logical relationship between the nodes and ports.

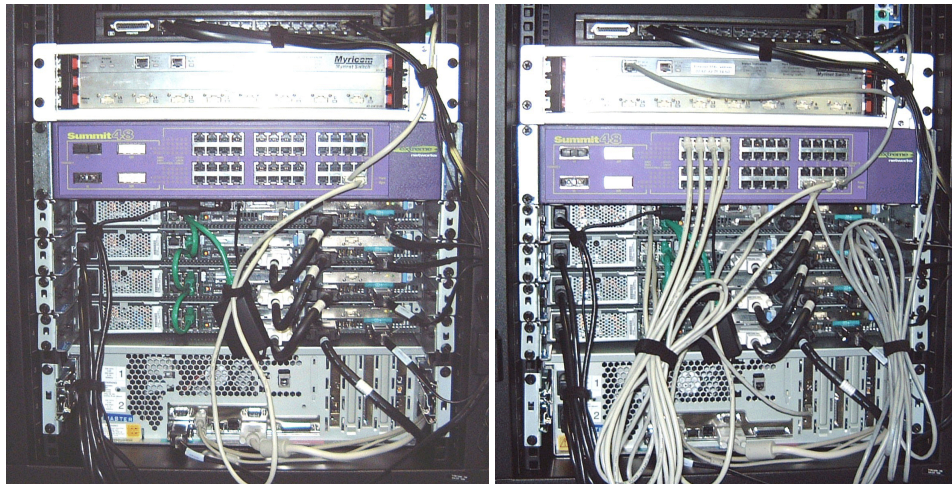


Figure 3-3 Terminal server cables (left) and FastEthernet cabling (right)

5. Connect the Gigabit Ethernet uplink from the management node and from any additional installation server nodes, if present, to the Ethernet switch (Figure 3-7 on page 41).

The following steps are optional and depend on your individual configuration and setup:

1. Connect the nodes that need external network connections (user and management nodes) to the external networks.
2. Interconnect the Ethernet switches from rack-to-rack (if applicable), preferably via Gigabit Ethernet.
3. Connect the Myrinet cables from the nodes to the switches.

4. If you are interconnecting Myrinet switches, it is recommended that you obtain a switch layout either directly from Myricom or from the IBM engineer involved with your cluster if services are included. Myrinet deployments requiring multiple switches can become very complex and require sophisticated cabling schemes to achieve full bisection bandwidth or optimum performance. See “Setting up the Myrinet switch” on page 221 for more information.
5. Next, review the cabling. It is recommended that you find someone else to do this.

Finally, all the cables should be laid out neatly, keeping in mind that you might have to recheck and/or exchange cables later when setting up the cluster management software and applications due to cabling errors or faulty cables. Use permanent cable ties for long cable runs that are not likely to be changed, and soft ties where cables may need to be moved for maintenance. Avoid dangling cables, but leave enough free cable length to permit replacement of any unit should a failure occur.

Figures 3-4 and 3-5 show the right side and rear view of a partially completed tall frame with Ethernet, MPN, and C2T cabling in place.



Figure 3-4 Power distribution units





*Figure 3-5 Cluster Ethernet, MPN, and C2T cabling*

### 3.3 Cables in our cluster

Figure 3-6 and Figure 3-7 on page 41 illustrate the back panel and cabling connections in our cluster used in the lab, configured with one Model 342 as a management and user node and eight Model 330s as compute nodes (only four nodes are shown in the diagram).

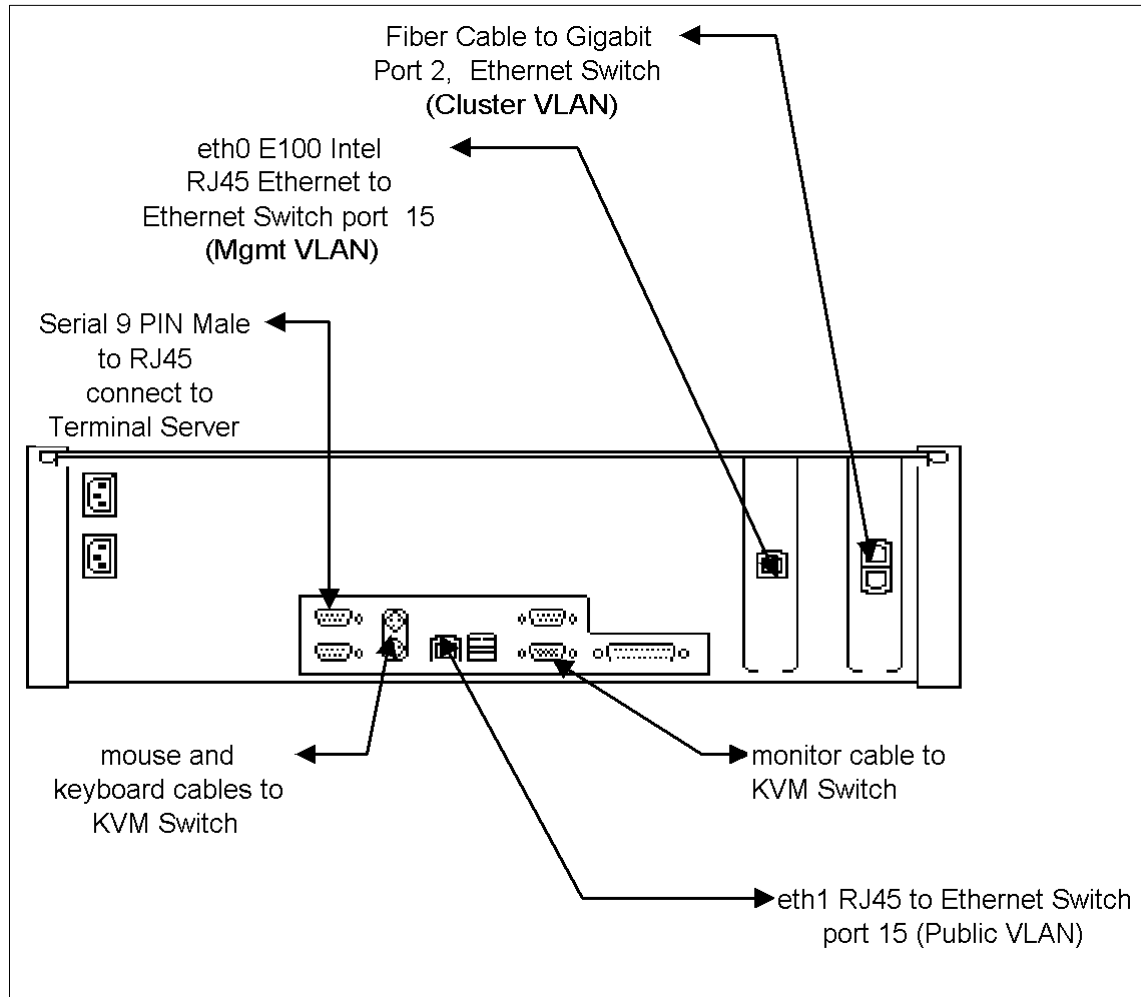


Figure 3-6 Cables on our master node (x342)



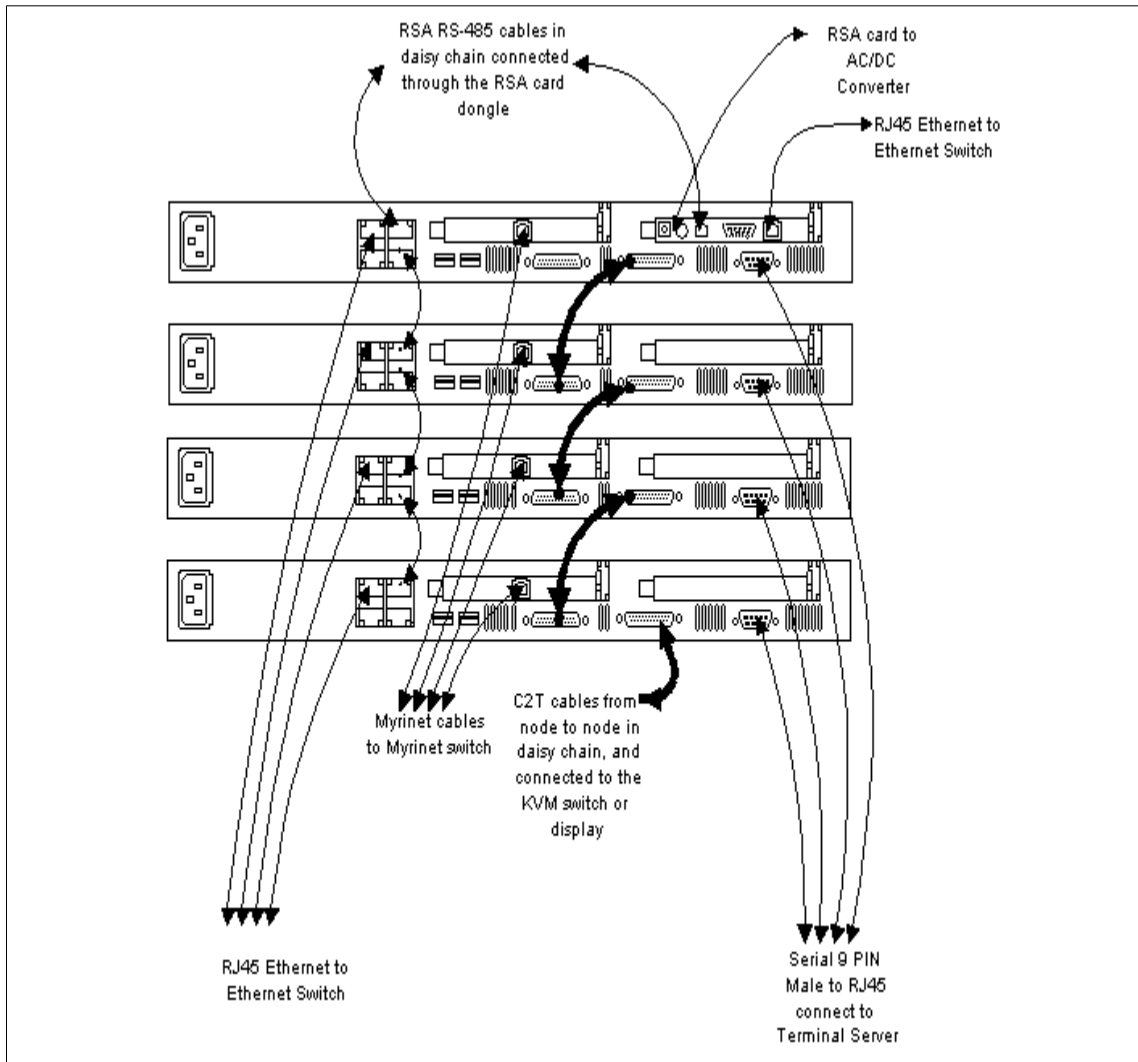


Figure 3-7 Cables on our compute nodes (x330)





## Management node installation

This chapter describes the installation of the management node. We discuss:

- ▶ Where to find help with the installation
- ▶ Steps to install Red Hat Linux
- ▶ How to apply the Red Hat errata
- ▶ How to install an updated driver for the Gigabit Ethernet Adapter

## 4.1 Resources to install Red Hat Linux

The IBM PC support Web site at <http://www.pc.ibm.com/support>, provides information on doing a Red Hat installation. The site is menu-driven, providing documentation based on the specified hardware and distribution level.

Enter your type/model number or select the following options, starting from the Browse menu:

1. Servers
2. Family (for our cluster, xSeries 342)
3. Machine type (8669)
4. Model (All)

At this point, the relevant part of your screen will look like Figure 4-1.

The screenshot shows the IBM PC support website interface for xSeries 342 8669 support. On the left is a navigation menu with the following items: PC support home, Online assistant, Discussion forums, Warranty lookup, Downloadable files, Hints and tips, Hot news, Online publications, Parts information, Product information, OS installation, and a Related links section containing Survey, Support phone list, Business Partner support, IBM PC Institute, and IBM Publications. The main content area is titled "xSeries 342 8669 support" and features a search bar with a "Go" button, a "Quick path to a product" section with a text input field for "Enter type-model" and a "Go" button, and a "Select your product" section with four numbered dropdown menus: 1. Servers, 2. xSeries 342, 3. 8669, and 4. All, each with a "Go" button. Below the search bar are radio buttons for "This product" (selected) and "All PC support", and a link for "Advanced search". Below the "Quick path to a product" section is a link for "Get help finding your machine-to-model".

Figure 4-1 xSeries 342 support

5. Select OS installation in the menu on the left-hand side.

6. Select a category: Linux.

The relevant part of your screen should now look like Figure 4-2.

Online publications  
Parts information  
Product information  
OS installation

**Related links:**  
Survey  
Support phone list  
Business Partner support  
IBM PC Institute  
IBM Publications Center  
Find a Business Partner

**Select your product**

① Servers

**Family**

② xSeries 342

**Machine type**

③ 8669

**Model**

④ All

**Operating system installation by category**

Linux

**Operating system installation - Linux by date** Documents 1 to 6 of 6

- [IBM eServer xSeries 342 - Installing Red Hat Linux v7.2](#) 2002-06-17
- [IBM eServer xSeries 342 - Installing SuSE Linux v7.3](#) 2002-06-17
- [IBM eServer xSeries 342 - Installing Red Hat Linux v7.1](#) **2002-06-14**
- [IBM eServer xSeries 342 - Installing TurboLinux Server v6.5](#) 2002-06-14
- [IBM eServer xSeries 342 - Installing Caldera OpenLinux v2.3](#) 2002-06-14
- [IBM eServer xSeries 342 - Installing SuSE Linux v7.1](#) 2002-06-14

Figure 4-2 IBM @server xSeries 342 - Installing Linux

7. Now select the document corresponding to your Linux distribution. In the lab we used Red Hat 7.3.

**Note:** Now that you have information on how to install Linux, the next step is to perform the actual installation. We recommend that you do a *custom* install.

## 4.2 Red Hat installation steps

The installation steps listed here correspond to the screens presented during a Red Hat installation.

**Tip:** More installation details for Red Hat 7.3 are at:

<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/>

The following are the Red Hat installation steps.

1. Select a language.

The language selected is used for the installation. It is independent of the language locale requested for the running kernel.

2. Select the keyboard and mouse.

Choose your keyboard and mouse types. In our lab, on x342, we used the following:

- Generic 105-key (Intl)
- Two-button mouse (PS/2)

3. Select the install option.

Select custom installation.

4. Create the following partitions.

Select **Disk Druid** to create partitions. The following list has our suggested partitioning scheme. The partitions are:

<b>/boot</b>	50 MB
<b>/install</b>	1 GB for xCAT, 2 GB per distribution, and 1 GB for updates per distribution
<b>/opt</b>	1 GB
<b>/var</b>	1 GB per 128 nodes
<b>/</b>	2 GB
<b>swap</b>	Double size of memory
<b>/tmp</b>	512 MB

Because they can grow unbounded, it is important to keep `/var` and `/tmp` in partitions separate from `/` [root]. If either is part of the root partition and the partition fills up (syslog and consver generate a lot of output), you would be unable to reboot the machine. `/`, `/usr`, `/home`, etc. can be kept separate or combined according to the taste of the administrator.

**Note:** It is easy to run out of space in /install. The Red Hat distribution grows with every release. For a Red Hat release that has been out there for a few months, there are a few hundred megabytes of updated packages available.

When you try to determine the size of the /install partition, allow for the following:

- ▶ Red Hat CD-ROMs. Allow about 650 MB per installation disk in the distribution. Some CD-ROMs have an SRPMS directory, which you can safely delete from the copy.
- ▶ Red Hat updates. Have a look on the Red Hat FTP server or on of the mirrors to see how much space they need and add some for future updates.
- ▶ Add 1 GB or more for use by xCAT and to have some extra free space, just in case.

Remember that 64-bit architecture machines have larger binaries so you may need more space in some partitions.

5. Select the bootloader.

Select **GRUB** (Grand Unified Bootloader), which is the default option.

6. Select a GRUB password.

We did not select a GRUB password.

7. Network configuration.

In our example there are three network adapters used in the management node. They attach to the public, management, and cluster networks. The management node uses the following interface configurations to connect to these networks:

- For eth0 (public VLAN)
  - Deselect DHCP (static addressing scheme).
  - Select **Activate on Boot**.
  - IP address is 192.168.42.1.
  - Netmask is 255.255.255.0.
  - Broadcast and network addresses are automatically generated.
  - Host name is masternode.clusters.com.
  - Gateway address is 192.168.42.254.

Both the IP and gateway addresses are site-specific.

- For eth1 (management VLAN)
  - Deselect DHCP (static addressing scheme).
  - Select **Activate on Boot**.
  - IP address is 10.1.0.1.
  - Netmask is 255.255.0.0.
  - Broadcast and network addresses are automatically generated.
  - Preserve host name and gateway address.
- For eth2 (cluster VLAN)
  - Deselect DHCP (static addressing scheme).
  - Select **Activate on Boot**.
  - IP address is 10.0.0.1.
  - Netmask is 255.255.0.0.
  - Broadcast and network addresses are automatically generated.
  - Preserve host name and gateway address.

**Note:** There is an optional fourth network, possibly high bandwidth-low latency, referred to as the IPC network. This network is used by message-passing applications. It is configured later during the xCAT post-install stage.

#### 8. Firewall configuration.

We use no firewall, but all nodes connected to the public VLAN may need to be behind a firewall. Certainly, any internal IP address (from the other three network VLANs) cannot have a firewall preventing communication between cluster nodes.

#### 9. Select language environments for runtime.

Select the language to be used as the default (locales for runtime Linux).

#### 10. Time zone.

You should always use UTC.

**Important:** Some problems were reported from users that did not use UTC. UTC is standard for all UNIX-like machines. The reason they did not check UTC was because RSA uses the CMOS clock and only understands local time. By using UTC, the SNMP alert timestamp of the MP clock will be the timestamp when the alert is received.

#### 11. Account configuration.

Set up the root password and, optionally, additional user accounts.



## 12. Authentication configuration.

If you want to use an external NIS server, and it does not support MD5 or shadow passwords, you need to uncheck the associated boxes here.

## 13. Select package groups.

These groups are brought up when custom installation (3., “Select the install option.” on page 46) is selected.

### – Recommended package groups

This is a recommended (brief) list of package groups to install.

- X Window System
- GNOME
- Network Support
- NFS File Server
- DNS Name Server
- Network Managed Workstation
- Utilities
- Legacy Applications
- Software Development
- Kernel Development

Also, check the Select individual packages box.

### – Additional packages

The next dialog allows you to see what packages have been selected and provides the opportunity to select additional packages.

**Tip:** If you know the name of the package, you can select the flat file display, which lists all available packages alphabetically.

Check to see that the following packages will also be installed. Use the flat file display to check for them and select them if necessary.

- dhcp
- expect
- kernel-smp
- pdksh
- uucp
- tftp

## 14. Video configuration.

This will auto-detect the card and video memory. Our lab has S3 Savage4.

#### 15. Install Packages.

Click **Next** and wait while the disks are formatted and the packages are installed. This takes a while.

#### 16. Boot disk creation.

You can skip this. The Red Hat 7.3 installation CD-ROM also works as a rescue disk.

#### 17. Monitor and customization.

This is where you complete the X configuration. The setup screens suggest monitor types, horizontal and vertical sync ranges, color depth, and screen resolution. Select a generic LCD display, 1024x768 resolution, and 16 bits color depth.

## 4.3 Post-installation steps

After Linux Red Hat 7.3 installation on the management node is completed, perform the following steps:

1. Copy the Red Hat install CD-ROMs to the management node to be used during the compute node installation.
2. Make sure you are running the package levels you need for your application and also apply the necessary updates found on the Red Hat errata Web site.
3. Download and install the necessary third-party drivers.

### 4.3.1 Copy Red Hat install CD-ROMs

For our example, the target is `/install/rh73`, based on the distribution level that we are using.

```
[root]# mkdir -p /install/rh73
```

Repeat the following commands for every installation CD (three CD-ROMs for Red Hat 7.3).

```
[root]# mount /mnt/cdrom
[root]# tar cf - -C /mnt/cdrom . | tar xvf - -C /install/rh73
[root]# umount /mnt/cdrom
```

**Note:** DMA issues with CD-ROM drives.

Some users have seen problems while reading from IDE CD-ROM drives in Netfinity and xSeries servers under Linux. The problems show up as DMA errors in the system log. Symptoms include very slow CD-ROM access and an inability to access the data on the disk. If you see this problem, we strongly recommend that you disable IDE DMA. To disable DMA you need to pass the following parameter to the kernel:

```
ide=nodma
```

To do this when you are installing Red Hat, type the following at the boot prompt after you boot from the CD-ROM:

```
linux ide=nodma
```

After the installation is complete, you need to add the parameter to your bootloader configuration. If your bootloader is GRUB, add the parameter to the line that loads the kernel in `/etc/grub.conf`, then:

```
kernel /vmlinuz-version ro root=/dev/sda1 ide=nodma
```

If your bootloader is LILO, add the following line to `/etc/lilo.conf`:

```
append="ide=nodma"
```

### 4.3.2 Install Red Hat errata

Red Hat will release updated packages for Red Hat Linux whenever a problem is found and fixed in their product. System administrators are advised to keep track of Red Hat updates and download and install new updates when they are released, especially if they fix a problem with security. The errata can be downloaded from:

<ftp://updates.redhat.com/7.3/en/os/>

Please check if there is a mirror site close to you:

<http://www.redhat.com/download/mirror.html>

The relevant errata are in `i386`, `i686`, and `noarch`.

Download the errata from Red Hat's FTP server and substitute the following with a host name and path of a mirror close to you:

```
[root]# cd /install  
[root]# wget -m ftp://updates.redhat.com/7.3/en/os/
```

Copy the errata to the xCAT post-install directory. Later the xCAT post-install procedure will install them on the cluster nodes from this directory:

```
[root]# mkdir -p /install/post/updates/rh73
[root]# cp updates.redhat.com/7.3/en/os/i386/*.rpm /install/post/updates/rh73
[root]# cp updates.redhat.com/7.3/en/os/i686/*.rpm /install/post/updates/rh73
[root]# cp updates.redhat.com/7.3/en/os/noarch/*.rpm /install/post/updates/rh73
```

When you try to update all these packages at once, there will be problems with conflicts and dependencies. This is usually because there are duplicate packages (Example 4-1). If a package is updated twice by Red Hat, both versions will be included, but you only need the latest version. The kernel and glibc packages are also included for i386 and i686 architectures, and you will need to keep only the i686 version.

We have included a small script (Example 4-2) to help you identify duplicate packages.

```
[root]# cd /install/post/updates/rh73
[root]# lsdup
```

We need to manually delete any extra packages, so that only one version of each package remains.

The script in Example 4-1 will display any duplicate packages in the current directory but it will not delete the duplicates. You need to list them and decide for yourself which packages to keep and which to delete.

*Example 4-1 Listing and filtering duplicates for Red Hat 7.3 at time of publication*

---

```
[root]# lsdup
kernel
[root]# ls kernel-*
kernel-2.4.18-4.i386.rpm
kernel-2.4.18-4.i686.rpm
kernel-bigmem-2.4.18-4.i686.rpm
kernel-BOOT-2.4.18-4.i386.rpm
kernel-debug-2.4.18-4.i686.rpm
kernel-doc-2.4.18-4.i386.rpm
kernel-smp-2.4.18-4.i686.rpm
kernel-source-2.4.18-4.i386.rpm
[root]# rm kernel-2.4.18-4.i386.rpm
```

---

*Example 4-2 lsdup script*

---

```
#!/bin/ksh
ls | sed 's/-[0-9]*\.*//g' | sort > /tmp/lsdup.all.$$
ls | sed 's/-[0-9]*\.*//g' | sort | uniq > /tmp/lsdup.uniq.$$
diff -y --suppress-common-lines /tmp/lsdup.all.$$ /tmp/lsdup.uniq.$$ |\
```

```
tr -d "[:blank:]\<" | sort | uniq
rm -f /tmp/lstdup.all.$$ /tmp/lstdup.uniq.$$
```

---

On the management node we have to install the updates by hand. The following command makes sure that you update all of the available packages except the kernel package, which needs special attention:

```
[root]# rpm -Fvh $(ls *.rpm | egrep -v '^(kernel-)')
```

Then install the new kernel (keeping the old one around for backup).

To install an updated kernel package, we use a different command to make sure that the old kernel remains installed in addition to the new kernel. This is done so that if the new kernel does not work on your system, you still have the old kernel, which you can select from the boot menu in order to recover the system.

```
[root]# rpm -ivh kernel-smp-2.4.18-4.i686.rpm
```

For Red Hat 7.3, the kernel package will add an entry to `grub.conf` but will not make it the default. You need to edit `grub.conf` and change the default line to:

```
default=0
```

**Note:** In previous versions of Red Hat, after installing an updated kernel package, you need to add an entry to `/etc/grub.conf` or `/etc/lilo.conf` for the updated kernel before you can use the new kernel. If your system uses an initial ramdisk, you also need to run `mkinitrd` to create the initial ramdisk for the new kernel. If your bootloader is LILO, do not forget to run the `lilo` command before rebooting.

Reboot the system with the new kernel:

```
[root]# reboot
```

**Tip:** The following command can be used to see which kernels are installed on the system, including the architecture that the kernel is compiled for.

```
[root]# rpm -q kernel kernel-smp --queryformat \
“%{NAME}-%{VERSION}-%{RELEASE}.%{ARCH}\n”
kernel-2.4.18-3.i386
kernel-smp-2.4.18-3.i686
kernel-smp-2.4.18-4.i686
```

### 4.3.3 Updating third party drivers

Systems using the e1000 device driver for Intel-based Gigabit Ethernet cards may experience problems. The symptoms seen are loss of network connectivity during heavy network traffic and problems in VLAN environments.

We recommend that you download the latest e1000 driver from the Intel Web site. As of this publication, the version of the driver in Red Hat 7.3 is 4.1.7. The version on intel.com is 4.2.17. You can download the latest driver from:

<http://support.intel.com/support/network/adapter/1000/software.htm>

The file name is e1000-4.2.17.tar.gz.

Before the driver can be installed, it needs to be compiled against your kernel. Make sure that you are running the correct kernel version.

```
[root]# uname -r
2.4.18-4smp
```

Also make sure that the source code for this kernel version is installed.

```
[root]# rpm -q kernel-source
kernel-source-2.4.18-3
```

This is the kernel source for the original Red Hat kernel. We need to remove it and install the updated kernel source instead:

```
[root]# rpm -e kernel-source
[root]# rpm -ivh /install/post/updates/rh73/kernel-source-2.4.18-4.i386.rpm
[root]# rpm -q kernel-source
kernel-source-2.4.18-4
```

**Tip:** It is safest to eliminate all but the latest kernel source. Make sure that only one kernel source is installed and that the kernel version running is the same as the kernel source version installed. Be sure that the symbolic link *linux-2.4* points to the proper kernel.

Now we are ready to compile the driver. The following command will compile the driver and create a binary RPM package that we use later to install the driver:

```
[root]# rpm -tb e1000-4.2.17.tar.gz
```

The RPM package is created in `/usr/src/redhat/RPMS/i386`:

```
[root]# rpm -ivh /usr/src/redhat/RPMS/i386/e1000-4.2.17-rh73.i386.rpm
```

Even with the latest driver, we have experienced problems during heavy traffic. You are most likely to notice this problem during the installation of the nodes, which generates a lot of NFS traffic. More information about this problem can be

found in the readme file included with the driver under the heading Known Issues. The readme file can be found in /usr/share/doc/e1000-4.2.17/.

The fix for this problem is to add the following line to /etc/modules.conf:


```
options e1000 RxIntDelay=0
```

Finally we need to reboot the system to activate the new driver:

```
[root]# reboot
```







# Management node configuration

In this chapter we explain how to configure the management node to run xCAT.  
We talk about:

- ▶ Installing xCAT
- ▶ xCAT configuration tables
- ▶ Management node services
  - Remote logging
  - SNMP, TFTP, NFS, NTP, SSH, DNS, and DHCP
  - Conserver
  - Network booting
  - Kickstart

## 5.1 Install xCAT

By default, xCAT is installed in /opt/xcat. Run the following command to unpack the distribution package:

```
[root]# tar xzpvf xcat-dist-1.1.0.tgz -C /opt/
```

Copy the xCAT profile to /etc/profile.d/. The profile adds the xCAT commands to the path and sets a variable XCATROOT.

```
[root]# cp /opt/xcat/samples/xcat.{sh,csh} /etc/profile.d
[root]# chmod 755 /etc/profile.d/xcat.{sh,csh}
```

xCAT provides a number of init scripts that are used to start and stop services, and these init scripts will run with a different environment, which means that XCATROOT is not defined. These scripts will look for XCATROOT in /etc/sysconfig/xcat. Run the following command to create this file:

```
[root]# echo "XCATROOT=/opt/xcat" > /etc/sysconfig/xcat
```

Log out and log in again to activate the xCAT profile.

**Note:** For this publication we used xCAT Version 1.1.0 to install the example cluster in our lab.

## 5.2 Populate tables

Most of the configuration files are tables that have one line for each node in the cluster. All files are plain text, so you can use your favorite text editor to create them. Many people roll their own scripts to generate the tables. Sample configuration files for xCAT can be found in /opt/xcat/samples/etc/.

The xCAT configuration files need to be located in /opt/xcat/etc. You have to create the directory first:

```
mkdir /opt/xcat/etc
```

We only discuss the tables that were used in the example cluster in our lab. Not every xCAT table is mandatory. Table 5-1 on page 59 is an overview of all the tables that xCAT can use.

Table 5-1 xCAT configuration tables overview

Table	Description
site.tab	This is the main xCAT configuration file. The table is mandatory, but not every field in this table is required.
nodelist.tab	Defines all nodes and groups. This table is mandatory.
noderes.tab	Installation resources for all nodes. This table is mandatory.
nodetype.tab	Defines the type for each node. This table is mandatory.
nodehm.tab	Defines the hardware management methods for each node. This table is mandatory.
mpa.tab	Defines the parameters for the MPA cards. This table is mandatory unless you have a cluster without IBM Advanced Systems Management.
mp.tab	Defines the Management Processor Network topology. This table is mandatory unless you have a cluster without IBM Advanced Systems Management.
apc.tab and apcp.tab	Defines the outlet used to power nodes or devices. Only needed if you have an APC MasterSwitch or MasterSwitch+ in your cluster.
cisco3500.tab	Defines the ports on the Cisco Ethernet switches for each node used for MAC address collection. Optional unless you use the Cisco switch for MAC address collection.
mac.tab	Lists the MAC address for every node. Generated by <b>getmacs</b> . This table is mandatory.
passwd.tab	Lists the user names and passwords used by xCAT scripts. This table is mandatory.

Table	Description
conserver.tab, rtel.tab, and tty.tab	Remote console configuration for any node. You will need at least one of these tables, but you can use a combination of these. Each console port can be listed either in <code>conserver.tab</code> or in the other tables, but not both.
nodemodel.tab	Used for remote flashing. Remote flashing is used at your own risk. This table is optional.

A full reference for all the tables may be found in Appendix B, “xCAT configuration tables” on page 185.

## 5.2.1 Site definition

`site.tab` (Example 5-1) contains information about the environment in which the cluster is to be run. xCAT’s system-wide settings are in this file.

### *Example 5-1 site.tab*

---

```
# site.tab for clusters.com

rsh          /usr/bin/ssh
rcp          /usr/bin/scp
gkhfile     /opt/xcat/etc/gkh
sshkeyver   1
tftpdirdir  /tftpboot
tftpxcatroot xcat
domain      clusters.com
dnssearch   clusters.com
nameservers 10.0.0.1
forwarders  NA
nets       10.0.0.0:255.255.0.0,10.1.0.0:255.255.0.0,10.2.0.0:255.255.0.0
dnsdir      /var/named
dnsallowq   10.0.0.0:255.255.0.0,10.1.0.0:255.255.0.0,10.2.0.0:255.255.0.0
domainaliasip NA
mxhosts     masternode
mailhosts   masternode
master      masternode
homefs      masternode:/home
localfs     masternode:/usr/local
pbshome     /var/spool/pbs
pbsprefix   /usr/local/pbs
pbsserver   masternode
scheduler   maui
```

```

xcatprefix      /opt/xcat
keyboard        us
timezone        US/Central
offutc          -6
mapperhost      NA
serialmac       0
serialbps       9600
snmpc           public
snmpd           masternode
poweralerts     Y
timeservers     masternode
logdays        7
installdir      /install
clustername     WOPR
dhcpver         2
dhcpconf        /etc/dhcpd.conf
dynamicr        eth2,ia32,10.9.0.1,255.255.0.0,10.9.1.1,10.9.254.254
usernodes       masternode
usermaster      NA
nisdomain       NA
nismaster       NA
nisslaves       NA
homelinks       NA
chagemin        0
chagemax        60
chagewarn       10
chageinactive   0
mpcliroot       /opt/xcat/lib/mpcli

```

---

Some of the fields are self-explanatory and many may be left at their default settings.

**Tip:** The Domain field must match the domain used in `/etc/hosts`. If you only want to use short host names you must still fill in this field; *cluster* would be one option.

## 5.2.2 Hosts file

If your cluster will run its own DNS service, which we recommend, xCAT can generate the DNS configuration automatically. To do this, you must list of all the host names and IP addresses that are used in your cluster in `/etc/hosts` (Example 5-2). It is then used to generate the configuration files for the DNS server.

*Example 5-2 /etc/hosts*

---

```
# Do not remove the following line, or various programs
```

```

# that require network functionality will fail.

127.0.0.1    localhost.localdomain localhost

# management node

10.0.0.1    masternode masternode.clusters.com
10.1.0.1    masternode-eth1 masternode-eth1.clusters.com
192.168.42.1 masternode-eth0 masternode-eth0.clusters.com

# compute nodes

10.0.1.1    node001        node001.clusters.com
10.0.1.2    node002        node002.clusters.com
10.0.1.3    node003        node003.clusters.com
10.0.1.4    node004        node004.clusters.com
10.0.1.5    node005        node005.clusters.com
10.0.1.6    node006        node006.clusters.com
10.0.1.7    node007        node007.clusters.com
10.0.1.8    node008        node008.clusters.com

# myrinet

10.2.1.1    node001-myri0  node001-myri0.clusters.com
10.2.1.2    node002-myri0  node002-myri0.clusters.com
10.2.1.3    node003-myri0  node003-myri0.clusters.com
10.2.1.4    node004-myri0  node004-myri0.clusters.com
10.2.1.5    node005-myri0  node005-myri0.clusters.com
10.2.1.6    node006-myri0  node006-myri0.clusters.com
10.2.1.7    node007-myri0  node007-myri0.clusters.com
10.2.1.8    node008-myri0  node008-myri0.clusters.com

# storage node

10.0.1.141  storage001     storage001.clusters.com
10.1.1.141  storage001-myri0 storage001-myri0.clusters.com

# rsa

10.1.1.101  rsa001         rsa001.clusters.com
10.1.1.102  rsa002         rsa002.clusters.com

# terminal servers

10.1.1.161  els001         els001.clusters.com
10.1.1.162  esp001         esp001.clusters.com

# cisco ethernet switches

```

```

10.1.0.241  cisco000          cisco000.clusters.com
10.1.1.241  cisco001          cisco001.clusters.com

# myrinet switch

10.1.0.201  myri001          myri001.clusters.com

# APC masterswitch

10.1.1.81   apc001           apc001.clusters.com

```

---

### 5.2.3 List of nodes and groups

This table (Example 5-3) must list all of the nodes and all of the devices in the cluster. Any device or node that needs to be managed by xCAT should be included in this table.

The first parameter in most xCAT commands is a **noderange**. It allows you to apply an operation over a set of nodes or devices, often in parallel. The **noderange** syntax is very powerful, but most of the time you will want to specify a group of nodes or devices. For this purpose, you can define groups in the node list. To do this, simply list the groups that each node or device is part of after the nodename, separated by commas.

#### *Example 5-3 nodelist.tab*

---

```

# nodelist.tab for clusters.com
#
# This table must list all the nodes in the cluster and defines the groups
# that each node is part of

node001    all,rack1,mpn1,compute,myri,pos0109
node002    all,rack1,mpn1,compute,myri,pos0110
node003    all,rack1,mpn1,compute,myri,pos0111
node004    all,rack1,mpn1,compute,myri,pos0112
node005    all,rack1,mpn1,compute,myri,pos0113
node006    all,rack1,mpn1,compute,myri,pos0114
node007    all,rack1,mpn1,compute,myri,pos0115
node008    all,rack1,mpn1,compute,myri,pos0116

storage001 all,rack1,mpn2,storage,myri,pos0107

rsa001     nan,mpa,rsa,pos0116
rsa002     nan,mpa,rsa,pos0107

els001     nan,ts,els,pos0132
esp001     nan,ts,esp,pos0119

```

cisco001	nan,ethernet,cisco,pos0117
myri001	nan,myrinet,pos0101
apc001	nan,power,apc,pos0118

---

You may choose the groups in any way that you feel will be useful. We recommend that you include at least the following groups:

<b>all</b>	All of the nodes (storage, user, compute) in the cluster. A number of commands, defined later in this chapter, assume that the <i>all</i> group contains all of the nodes.
<b>rackN</b>	Contains all of the nodes in a particular rack.
<b>mpnN</b>	All of the nodes connected to a particular MPN.
<b>compute</b>	All of the compute nodes.
<b>storage</b>	All of the storage nodes.
<b>user</b>	Any user node.
<b>stage</b>	Any staging nodes for installation.
<b>myri</b>	All of the nodes on the Myrinet.
<b>posRRUU</b>	The physical location of the device in the cluster by rack (RR) and unit (UU).
<b>nan</b>	Any device that is not a node.
<b>mpa</b>	Any device that is an MPA (for example, RSA, ASMA).
<b>rsa</b>	Any device that is an RSA card.
<b>ts</b>	Any device that is a terminal server.
<b>els</b>	Any device that is an ELS terminal server.
<b>ethernet</b>	Any device that is a managed Ethernet switch.
<b>cisco</b>	Any Cisco Ethernet switch.
<b>power</b>	Any device that is a managed PDU.
<b>apc</b>	Any power supply that is an APC.

## 5.2.4 Installation resources

noderes.tab, as shown in Example 5-4 on page 65, contains all of the parameters required for the remote unattended installation of the nodes.

In a small cluster, such as the example cluster in our lab, we normally need only a single set of resources for the entire cluster. For large cluster installations you



may need to have multiple sets of resources. For example, a large cluster will have staging nodes. You will need to define a set of resources for every group of nodes that install from the same staging node. The staging nodes will also require a set of resources of their own, with the install role field set to Y.

*Example 5-4 noderes.tab*

---

```
# noderes.tab for clusters.com
#
# tftp,nfs,install_dir,serial,nis,install_role,acct,gm,pbs,access,gpfs,ksdevice

compute  masternode,masternode,/install,0,N,N,N,Y,N,N,N,NA
storage  masternode,masternode,/install,0,N,N,N,Y,N,N,N,NA
```

---

## 5.2.5 Node types

This file (Example 5-5) specifies the Kickstart file that will be used to install each node. Since the Kickstart file contains the post-installation script, this will affect the node's configuration. The Kickstart file is generated from the template file `/opt/xcat/ks73/nodetype.kstmp` with a number of values automatically generated from the xCAT tables.

If you have any special requirements you may need to modify the Kickstart template files or create your own additional template files (see 6.4.1, "Creating a template file" on page 107). You may need to change the partitioning scheme or the post-install script for certain node types.

*Example 5-5 nodetype.tab*

---

```
node001  compute73
node002  compute73
node003  compute73
node004  compute73
node005  compute73
node006  compute73
node007  compute73
node008  compute73

storage001  storage73
```

---

## 5.2.6 Node hardware management

One of the key features of xCAT is its hardware management support. The node hardware management table details which methods of hardware management are available to xCAT. See Example 5-6 on page 66.

### Example 5-6 *nodehm.tab*

---

```
# nodehm.tab for clusters.com
#
# This table defines the hardware management method used for each node
#
# power,reset,cad,vitals,inv,cons,rvid,eventlogs,getmacs,netboot,eth0,gcons,
# serialbios

node001 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node002 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node003 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node004 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node005 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node006 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node007 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y
node008 mp,mp,mp,mp,mp,mp,conserver,rcons,mp,cisco3500,pxe,e100,vnc,Y

rsa001 apc,apc,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA
rsa002 apc,apc,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA

els001 apc,apc,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA
esp001 apc,apc,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA,NA
```

---

We use the management processor for power control, vital statistics, inventory, and to read the event logs. We use a Cisco 3500-series switch to collect MAC addresses. The APC MasterSwitch is used to control devices that do not have internal power control.

Please see Appendix B, “xCAT configuration tables” on page 185, for information on all the possible values.

## 5.2.7 MPN topology

In order to utilize the MPN, xCAT needs to know its topology. This is configured in the MP table (Example 5-7 on page 66).

### Example 5-7 *mp.tab*

---

```
# mp.tab for clusters.com
#
# This table describes the topology of the Management Processor Network. This
# enables xcat to contact the management processors via the mpa connected to
# the same chain.
#
# mpa name of the mpa connected to this node
# name internal name of the mp adapter (*)
#
# (*) this field should be NA if the mpa is the primary management
```

```

#      adapter for that node
#
#      mpa,name

node001  rsa001,node001
node002  rsa001,node002
node003  rsa001,node003
node004  rsa001,node004
node005  rsa001,node005
node006  rsa001,node006
node007  rsa001,node007
node008  rsa001,node008

storage001rsa002,NA

```

---

When xCAT wants to send a command to the MP in a certain node, xCAT connects via Ethernet to the MPA card that is part of the same MPN. Then it uses the MPTN to connect to the MP.

## 5.2.8 MPA configuration

The table shown in Example 5-8 is used to configure the MPA cards and to select the interface (`telnet` or `mpcli`) appropriate to for the type of MPA.

*Example 5-8 mpa.tab*

---

```

# mpa.tab for clusters.com
#
# This table lists the Management Processor Adapters in this cluster
#
# type          asma, rsa
# name          internal name (*)
# number        internal number, unique and > 10000
# command       telnet, mpcli
# reset         http (asma only), mpcli, NA
# rvid          telnet (asma only), NA
# dhcp          Y/N (rsa only)
# gateway       default gateway IP address or NA
#
# (*) internal name should be the node name if the mpa is the
#     primary management adapter for that node
#
#      type,name,number,command,reset,rvid,dhcp,gateway

rsa001  rsa,rsa001,10001,mpcli,mpcli,NA,N,NA
rsa002  rsa,storage001,10002,mpcli,mpcli,NA,N,NA

```

---

## 5.2.9 Power control with APC MasterSwitch

The APC table (Example 5-9) indicates the outlet of a particular APC that controls the power for a particular device.

*Example 5-9 apc.tab*

---

```
# apc.tab for clusters.com
#
# This table defines the outlet and the APC MasterSwitch that can be used
# to manage the power on a particular node or device
#
# hostname      hostname of the APC MasterSwitch
# outlet        outlet connected to the device (1-8)
#
#           hostname,outlet

rsa001    apc001,1
rsa002    apc001,2
esp001    apc001,7
els001    apc001,8
```

---

### 5.2.10 MAC address collection using Cisco 3500-series

Before the cluster can be configured and installed, you need to know the MAC address and the physical location of every node in the cluster. You can collect the MAC addresses by hand, but even for relatively small clusters, this process is time consuming and prone to mistakes. The procedure can be automated using the terminal servers or the Ethernet switch.

In both cases, the principal is the same. The idea is that you always connect the cable from node 1 to port 1 on the first device, and repeat this for all nodes. In our example we used the Cisco 3524XL in our cluster to do the MAC address collection. Our recommendation is that, if possible, you always use the Ethernet switch for MAC address collection because it is faster and more reliable. If you need to use the terminal server for MAC address collection, then you do not need this table.

All Ethernet switches keep a MAC address table that allows them to forward traffic for a particular MAC address to the port that is connected to that address. The MAC address collection procedure will then download the MAC address table from the switch and use it to find the MAC address that belongs to each of the ports.

You will need to provide a table (Example 5-10 on page 69) that associates all the nodes with a port on a particular Ethernet switch.

*Example 5-10 cisco3500.tab*

---

```
# cisco3500.tab for clusters.com
```

```
node001    cisco001,1
node002    cisco001,2
node003    cisco001,3
node004    cisco001,4
node005    cisco001,5
node006    cisco001,6
node007    cisco001,7
node008    cisco001,8
```

```
storage001 cisco001,9
```

---

## 5.2.11 Console server configuration

The conserver console server allows simultaneous access to remote system consoles by multiple users. A transcript of each session is also logged to a file in `/var/log/consoles`, even when no users have opened a remote console.

For each console server in your cluster, you will need to provide a configuration file. The configuration file (see Example 5-11) specifies which remote consoles are available and how to connect to them. Typically you will run only a single console server on your management node, which in turn manages all the terminal servers in your cluster. In larger installations, you may choose to install additional console servers, for example, on the user or staging nodes.

In our example we used Equinox ELS-16 terminal servers. To configure a conserver for use with an ELS, you need to write a line for each node that looks like this:

```
nodename-con:!hostname:port:&:
```

Replace *hostname* with the host name of the terminal server, prefixed with an exclamation mark. For an ELS, the port numbers are TCP ports 3001 to 3016, where 3001 corresponds to physical port 1 and 3016 corresponds to physical port 16. For other types of terminal servers, please refer to the documentation included with the device. The ampersand simply means that conserver should use the node name as the file name of the log file for this console.

To configure the console server using an Equinox ESP, see “Equinox ESP Terminal Servers” on page 212.

*Example 5-11 conserver.cf*

---

```
#
# $Id: conserver.cf,v 1.3 1999-01-25 14:38:19-08 bryan Exp $
```

```

#
# The character '&' in logfile names are substituted with the console
# name. Any logfile name that doesn't begin with a '/' has LOGDIR
# prepended to it. So, most consoles will just have a '&' as the logfile
# name which causes /var/log/consoles/consolename to be used.
#
LOGDIR=/var/log/consoles
#
# list of consoles we serve
# name : tty[@host] : baud[parity] : logfile : mark-interval[m|h|d]
# name : !host : port : logfile : mark-interval[m|h|d]
# name : |command : : logfile : mark-interval[m|h|d]
#
node001-con:!els001:3001:&:
node002-con:!els001:3002:&:
node003-con:!els001:3003:&:
node004-con:!els001:3004:&:
node005-con:!els001:3005:&:
node006-con:!els001:3006:&:
node007-con:!els001:3007:&:
node008-con:!els001:3008:&:
%%
#
# list of clients we allow
# {trusted|allowed|rejected} : machines
#
trusted: 127.0.0.1

```

---

Even if you have a single conserver on the master node, you need to provide `conserver.tab` (Example 5-12), which is used by the console commands to locate the console server and the console name they need to connect to.

#### *Example 5-12 `conserver.tab`*

---

```

# conserver.tab for clusters.com
#
# This table defines the relationship between nodes and console servers. This
# example uses only one conserver on the masternode.
#
# conserver    hostname of the console server
# console      name of the console, corresponds to conserver.cf
#
#           conserver,console

node001  masternode,node001-con
node002  masternode,node002-con
node003  masternode,node003-con
node004  masternode,node004-con
node005  masternode,node005-con

```

```
node006  masternode,node006-con
node007  masternode,node007-con
node008  masternode,node008-con
```

---

If you chose not to use conserver, you need to provide `rtel.tab` or `tty.tab` to configure your terminal servers.

## 5.2.12 Password table

Example 5-13 defines the passwords that xCAT needs to access certain devices. It also defines the root password that is set on all nodes.

*Example 5-13 passwd.tab*

---

```
# passwd.tab for clusters.com
#
# defines the passwords that xCAT needs to access certain devices and the
# root password it needs to set on all nodes

rootpw  netfinity

# telnet and enable secret for Cisco IOS
cisco   cisco

# userid and password for the mpa and mp
asmauser  USERID
asmapass  PASSWORD
```

---

The 0 in PASSWORD is the digit zero.

**Tip:** If you want to change the default user ID and password for the MPA and MP, you will also need to modify `/opt/xcats/stage/stage3.tmpl`. This is because the password that stage 3 will set is hard-coded in this file.

## 5.3 Configure management node services

You are now ready to start the configuration of the management node. When all the following steps complete successfully, you can begin the installation of the cluster itself.

### 5.3.1 Turn off services you do not want

By default, Red Hat Linux enables a number of services that are not usually needed on the management node. For security and serviceability reasons, we

recommend that you disable any services that are not needed using the **chkconfig** command included with Red Hat Linux.

The following command shows the services that are currently configured to start on boot:

```
[root]# /sbin/chkconfig --list | grep ':on'
```

Use the following command to disable the services that are not needed. The list includes all the services that we disabled in our example cluster. You should modify the list to your own requirements.

```
[root]# LIST="kudzu apmd autofs iptables ipchains rawdevices lpd rhnsd"  
[root]# for SERVICE in $LIST; do /sbin/chkconfig --del ${SERVICE}; done
```

**Tip:** The list of services that are not required will vary from version to version and also depends on the packages you have installed. We recommend that you disable all of the following services.

- ▶ autofs
- ▶ linuxconf
- ▶ reconfig
- ▶ isdn
- ▶ pppoe
- ▶ iptables
- ▶ ipchains
- ▶ apmd
- ▶ pcmcia
- ▶ rawdevices
- ▶ lpd
- ▶ kudzu
- ▶ pxe
- ▶ rhnsd
- ▶ wine
- ▶ httpd
- ▶ identd

### 5.3.2 Configure system logging

The nodes in the cluster will send all their log entries to the master node. To enable the master node to accept remote log connections, the syslog configuration needs to be changed:

```
[root]# vi /etc/sysconfig/syslog
```

Change the following line:

```
SYSLOGD_OPTIONS="-m 0"
```



To:

```
SYSLOGD_OPTIONS="-m 0 -r"
```

**Tip:** It is handy to see the log messages scrolling by on a virtual terminal. After adding the following line to `syslog.conf`, you can see the last screen full of log messages with `Alt+F12`:

```
[root]# vi /etc/syslog.conf
```

Add the following line:

```
*.info;mail.none;news.none;authpriv.none;cron.none    /dev/tty12
```

Restart the syslog daemon so that these changes take effect.

```
[root]# service syslog reload
```

### 5.3.3 Configure SNMP

SNMP is used on the master node to receive alerts, which are called *traps* in SNMP. We also configure the SNMP daemon to send e-mail to the system administrators.

First, check that the `snmp` packages are installed. Install them from the Red Hat CD-ROM if required.

```
[root]# rpm -q ucd-snmp ucd-snmp-utils
```

Included with xCAT is an init script and a configuration file for SNMP. It configures SNMP to send e-mail to an e-mail alias called *alerts*, which you need to create.

First, copy the init script and configuration file that is provided with xCAT.

```
[root]# cp /opt/xcat/rc.d/snmptrapd /etc/rc.d/init.d
[root]# cp /opt/xcat/samples/etc/snmptrapd.conf /opt/xcat/etc
```

Then create the e-mail alias “alerts” and add the e-mail addresses of all the administrators that want to receive alerts. You need to add a line like the following in `/etc/aliases`, using your own system administrators’ e-mail addresses, of course.

```
alerts: be@clusters.com,sdenham@clusters.com,turcksin@clusters.com
```

After you add an alias, you need to rebuild the mail alias database with the following command:

```
[root]# newaliases
```

Finally, enable and start the snmptrapd service:

```
[root]# chkconfig --add snmptrapd
[root]# service snmptrapd start
```

### 5.3.4 Configure TFTP

PXE network boot clients use TFTP to transfer the kernel and all the other files they need from the boot server. In a small cluster, like the example cluster in our lab, the master node runs the TFTP server. On larger installations using staging nodes, each staging node will also run a TFTP server.

To install and start the TFTP server provided with xCAT, run the following commands:

```
[root]# rpm -ivh /opt/xcat/post/rpm73/atftp-0.6-1.i386.rpm
[root]# chkconfig --add atftpd
[root]# service atftpd start
```

Make sure that the TFTP server is working before you continue, using the following procedure:

```
[root]# mkdir /tftpboot
[root]# echo "Hello, world" >/tftpboot/test
[root]# tftp masternode
tftp> get test
tftp> quit
[root]# cat test
Hello, world!
[root]# rm test /tftpboot/test
```

**Note:** We use ATFTP, provided with xCAT, because the original TFTP server included in Red Hat was not suitable for use in clusters; ATFTP is scalable to large clusters. Most TFTP servers are started by the inet daemon, but ATFTP can run stand-alone.

In our experience, we have found that ATFTP sometimes stops running and needs to be restarted. If this happens to you, you may want to try to use the TFTP server included in Red Hat 7.2 and higher. The new Red Hat TFTP server is based on tftp-hpa by H. Peter Anvin of SYSLinux and PXELinux fame.

### 5.3.5 Configure NFS

NFS is used to install Red Hat on all of the cluster nodes over the network.

You need to export a number of file systems from the master node using NFS.

- ▶ For the network installation of Red Hat on all of the nodes in the cluster, the Red Hat CD-ROMs needs to be exported. In “Copy Red Hat install CD-ROMs” on page 50, the Red Hat CD-ROMs were copied to `/install/rh73/`. The `/install/` directory also contains the post-install directory and the Red Hat errata.
- ▶ All of the cluster nodes will mount the xCAT directory from the master node.
- ▶ The cluster nodes can also mount `/home` and `/usr/local` over NFS for the convenience of users. This is optional. See `site.tab` variables `localfs` and `homefs` in Example 5-1 on page 60.

Example 5-14 shows added lines to `/etc/exports`.

*Example 5-14 /etc/exports*

---

```
/install    *(ro,no_root_squash)
/opt/xcat   10.0.0.0/16(ro,no_root_squash)
/usr/local  10.0.0.0/16(ro,no_root_squash)
/home      10.0.0.0/16(rw,no_root_squash)
```

---

The example shown allows world access for `/install/` but limits access to nodes on the cluster VLAN for the other exports for better security. This is just an example and you should consider the appropriate security settings for your own configuration.

Run the following commands to enable and start the NFS service.

```
[root]# chkconfig nfs on
[root]# service nfs start
```

The `exportfs` command can be used to verify that the file systems are now exported via NFS.

```
[root]# exportfs
/usr/local    10.0.0.0/16
/opt/xcat     10.0.0.0/16
/home        10.0.0.0/16
/install     world
```

### 5.3.6 Configure NTP

It is important that all the nodes in the cluster have synchronized time. If you have a reliable time source you should use it, but even when this is not available you should at least use the NTP service to make sure that all of the nodes have their clocks set to the same time, even if it is not quite the correct time.

**Tip:** Most networks have time servers available that you can use as a reliable source of time. To use this service, find out the host name or IP address of the time server(s) on your network, and add them to `/etc/ntp.conf`.

```
restrict hostname mask 255.255.255.0 nomodify notrap noquery
server hostname
```

During the installation, all the nodes will configure themselves to synchronize the time with the management node. This means you must enable the NTP service on the management node and allow the clients to synchronize with the master node.

The default `ntp.conf` in Red Hat 7.3 restricts access to the NTP service. To allow clients to synchronize time with the master node, you can either lift the restriction completely or allow access for certain networks only. In our example, we added the following lines to `/etc/ntp.conf`, one for each of the subnets in the cluster:

```
[root]# vi /etc/ntp.conf
restrict 10.0.0.0 mask 255.255.0.0 notrust nomodify notrap
restrict 10.1.0.0 mask 255.255.0.0 notrust nomodify notrap
restrict 192.168.42.0 mask 255.255.255.0 notrust nomodify notrap
```

Use the following commands to enable and start the NTP service:

```
[root]# chkconfig ntpd on
[root]# service ntpd start
```

**Tip:** If you want to test your timeserver after starting or restarting it, be aware that it can take up to a few minutes before the server is ready to allow synchronization. Keep an eye on the system log for a status change message from `ntpd` to indicate when it is ready:

```
masternode ntpd[20522]: kernel time discipline status change 41
```

### 5.3.7 Configure SSH

The cluster administrator needs to have full control of the complete cluster from a single point of entry. Also, the users of the cluster need to be able to start jobs on many nodes seamlessly. Both `rsh` and `ssh` can do this. `ssh` is the better choice for this task because it is more secure.

The `gensshkeys` script creates a private/public key pair with an empty passphrase for root. It also creates a default SSH configuration to make sure that root can open secure shell sessions to and from the master node and every other node in the cluster. These files are in `/root/.ssh`.

The files in `/root/.ssh` are copied to `/install/post/.ssh` so that they can be distributed to all the nodes in the cluster during the post-install operation. This allows the administrator to run commands on all nodes in the cluster simultaneously without needing to type the password for every one.

```
[root]# gensshkeys root
```

Enable SSH.

```
[root]# chkconfig --add sshd
[root]# service sshd start
```

### 5.3.8 Configure the console server

Conserver can be downloaded from <http://www.conserver.com/>. The Web site also contains more information about conserver.

xCAT provides a prebuilt conserver binary and init script.

Copy the conserver init script and start the service.

```
[root]# cp /opt/xcat/rc.d/conserver /etc/rc.d/init.d
[root]# chkconfig conserver on
[root]# service conserver start
```

### 5.3.9 Configure DNS

xCAT can generate the DNS configuration files automatically. In order to do this, you must make sure that the IP address of the master node is the first address in the name servers field in `site.tab` (Example 5-1 on page 60).

To generate the DNS configuration, make sure that you have a correct and complete host file (Example 5-2 on page 61) and then run:

```
[root]# makedns
```

**Note:** The `makedns` command looks at the `nameservers` parameter in `site.tab` and does the following:

- ▶ If one of the node IPs matches the first field in name servers, then it is the master.
- ▶ If one of the node IPs matches any field other than the first, it does a zone transfer from the first and designates it as a secondary name server. It is used in environments where the DNS servers are managed elsewhere, but a local DNS server is desired to off-load local requests.
- ▶ If none match it becomes a client.

The **makedns** command will generate the configuration and restart the DNS server. At this point, make sure to check in the log file (remember to use Alt+F12 if you enabled it) to make sure that the DNS server did not log any error messages and that all zone files were loaded correctly. You may also want to inspect the zone files in `/var/named/` to ensure that everything is correct. A small typo in `site.tab` can cause problems here.

After DNS is started, test that the name resolution works using the **host** command. Make sure that both forward and reverse queries work and that the query fails promptly if you specify nonexistent host names or IP addresses.

### 5.3.10 Configure DHCP

To prevent the DHCP server from responding to DHCP requests on the public VLAN, you should configure it to listen only on the cluster and management VLAN. To do this, edit `/etc/sysconfig/dhcpd` and make sure it contains the following line:

```
[root]# vi /etc/sysconfig/dhcpd
DHCPDARGS="eth1 eth2"
```

The **makedhcp** command is used to generate and modify the DHCP configuration file. It also stops and starts the DHCP server for you. The following command generates the initial DHCP configuration file, `/etc/dhcpd.conf`. Be careful when you run this, because it will overwrite the file if it already exists.

```
[root]# makedhcp --new
```

**Tip:** Some devices can be configured using BOOTP or DHCP. To support this, the DHCP server needs to know the MAC addresses of those devices. Get the MAC address for each device (usually on a label on the casing) and list each in `/opt/xcat/etc/mac.tab`.

```
[root]# vi /opt/xcat/etc/mac.tab
apc001    00:C0:B7:A3:97:C5
myri001   00:60:DD:7F:35:29
```

The following command will then add the DHCP/BOOTP entries to `/etc/dhcpd.conf` and restart the DHCP server.

```
[root]# makedhcp --allmac
```

After this, wait a while and watch the system log to see if the device can successfully request an IP address. If the DHCP server has already allocated an IP address from the dynamic range to any device before, you may need to reset the device to force it to request the right address. Note that everything in xCAT is a node, so for `--allmac` to work, every xCAT resource must be defined in `nodelist.tab`.

## 5.4 Final preparation

Before we can begin the installation of the nodes, some final preparation is needed. You need to copy boot images to the `/tftpboot` directory and create the Kickstart scripts for the automatic installation. xCAT provides scripts to do the hard work for you.

### 5.4.1 Prepare the boot files for stages 2 and 3

During stages 2 and 3, the nodes boot from the network using PXE and load a small Linux kernel and a boot image. The **mkstage** command in `/opt/xcat/stage` will create the boot image and copy all of the required files to the correct places.

```
[root]# cd /opt/xcat/stage
[root]# ./mkstage
```

**Note:** The **mkstage** command needs to be run from the `/opt/xcat/stage` directory. There is also a **mkstage** command in `/opt/ks73` and the Kickstart directories for other Red Hat versions that have a different purpose.

## 5.4.2 Prepare the Kickstart files

As previously stated in “Node types” on page 65, the Kickstart files are generated from templates. This needs to be done before you can install Linux on any node in the cluster.

```
[root]# cd /opt/xcat/ks73
```

For each type of node defined in `nodetyp.tab`, create another Kickstart template. We start from a copy of the compute node, and then customize this file.

```
[root]# cp compute73.kstmp storage73.kstmp
[root]# vi storage73.kstmp
[root]# ./mkks
```

It is likely that you will customize the Kickstart scripts during and after the installation. Be sure to rerun `mkks` every time you make a modification.

**Note:** There is a directory with Kickstart templates and an `mkks` command for every version of Red Hat that is supported. If you need to support multiple versions of Red Hat in the cluster, you will need to run the `mkks` command in each of the directories that correspond to the Red Hat versions that you need.

Also, you are required to run the `mkks` command every time you change an xCAT table.

## 5.4.3 Prepare the post installation directory structure

During the post-installation phase, certain files are needed in `/install/post`. The required directory structure needs to be copied from `/opt/xcat/post`.

```
cp -vr /opt/xcat/post/ /install/
```

### Additional RPM packages

If you have an RPM package that you want to install on all of the nodes, you should copy it to `/install/post/rpm73/`. All of the RPM packages in this directory are automatically installed during post-install.

### GPFS

If you want to use GPFS on your cluster, the GPFS software packages need to be copied to `/install/post/gpfs/`. You also need to enable GPFS support in `nodes.tab` (see “Installation resources” on page 64).



## Red Hat errata

In “Install Red Hat errata” on page 51, the updated RPM packages have already been copied to `/install/post/updates/rh73/`.

## Custom kernels

For Red Hat 7.3, we recommend that you use the Red Hat kernel. For previous versions of Red Hat, <http://x-cat.org/download/xcat/> has pre-built custom kernels. If you have special requirements you can also build a customized version of the kernel. These kernel packages need to be placed in `/install/post/kernel/`. Then edit the `KERNELVER` variable in the Kickstart script to select the kernel version you want to install.

The `KERNELVER` variable must also be set for Red Hat updated kernels. Note that the post-installation process does not update the kernel with other updated RPMs. It only sets the `KERNELVER` variable to match the version of the updated kernel (see <http://updates.redhat.com/>).





## Cluster installation

This chapter describes the steps required to prepare your cluster nodes for use. At this point you have a functional management node, and xCAT has been installed and configured. You would also use some or all of these steps to add additional nodes or racks of nodes to your existing cluster.

In this chapter the following topics are described:

- ▶ Configuring the network switch
- ▶ Preparing the Management Supervisor Network
- ▶ Setting up your terminal servers
- ▶ Updating the cluster node BIOS and BIOS settings
- ▶ Collecting hardware MAC address information
- ▶ Installing an operating system on each node

## 6.1 Stage 1: Hardware setup

This stage involves configuring the hardware that you have successfully connected together. This hardware provides the infrastructure upon which the cluster is built, and enables automated installation of the nodes. The first thing we need is a network, so the Ethernet switch is configured first, followed by the Management Processor Adapters and then the terminal servers.

### 6.1.1 Network switch setup

To configure the network we need to assign an IP address to the control function of the switch, and configure the ports and VLANs. In our lab, we used a Cisco 3524 switch and these configuration steps:

1. The initial configuration of the switch is done via the serial port management connection. Connect a Cisco Modular Adapter and a Cisco cable from the serial port of your switch to a Linux system (using `cu`), a Windows system (using a terminal emulator), or a real serial terminal. Configure for 9600 baud, 8 bits, no parity, and 1 stop bit.

2. Create an empty file in `/tftpboot` to receive the configuration, and open the permissions on it so it can be written by a TFTP transaction.

```
[root]# touch /tftpboot/cisco001.config
[root]# chmod a+w /tftpboot/cisco001.config
```

3. Configure the IP address and access passwords on the switch, and transfer the basic configuration to the master node, as shown in Example 6-1. Be sure that the passwords you specify here match what you specified in the `passwd.tab` file.

#### *Example 6-1 Cisco 3524XL initial configuration dialog*

---

```
--- System Configuration Dialog ---
```

```
At any point you may enter a question mark '?' for help.
Use ctrl-c to abort configuration dialog at any prompt.
Default settings are in square brackets '[]'.
```

```
Continue with configuration dialog? [yes/no]: yes
Enter IP address: 10.1.1.241
Enter IP netmask: 255.255.0.0
Would you like to enter a default gateway address? [yes]: no
Enter host name [Switch]: cisco001
```

```
The enable secret is a one-way cryptographic secret used
instead of the enable password when it exists.
```

```
Enter enable secret: cisco
```

```
Would you like to configure a Telnet password? [yes]: yes
Enter Telnet password: cisco
Would you like to enable as a cluster command switch? [yes/no]: no
```

The following configuration command script was created:

```
ip subnet-zero
interface VLAN1
ip address 10.1.1.241 255.255.0.0
hostname cisco001
enable secret 5 $1$HWuA$5FJfDhgM73Cy2U0ca5Zca.
line vty 0 15
password cisco
snmp community private rw
snmp community public ro
!
end
```

```
Use this configuration? [yes/no]: yes
Building configuration...
[OK]
Use the enabled mode 'configure' command to modify this configuration.
```

Press RETURN to get started.

```
cisco001>en
Password:
cisco001#copy running-config tftp:
Address or name of remote host []? 10.1.0.1
Destination filename [cisco001-config]?
!!
1404 bytes copied in 1.492 secs (1404 bytes/sec)
```

---

4. Edit the file /tftpboot/cisco001-config to configure the ports and define the VLANs. The initial file transferred from the switch is shown in Example 6-2 on page 86, and the edited file is shown in Example 6-3 on page 87, with the additions we made shown in bold text. These changes include defining the cluster, management, and cluster VLANs, and configuring the ports for optimum performance. In our lab, ports 1–8 connect to the compute nodes and port 9 connects to the storage node. Gigabit port 1 connects to the Gigabit Ethernet switch that would tie multiple frames together, and Gigabit port 2 connects to our master node. These are assigned to the cluster VLAN (2). Ports 10–20 connect to management equipment (the RSA cards, the APC switch, the terminal servers), and are assigned to the management VLAN (1). Ports 21–24 attach to an external server and the outside world, and are assigned to the public VLAN (3).

**Tip:** By default, all ports on the Cisco switch run the spanning-tree protocol that analyzes the LAN configuration for loops. This introduces up to a 30 second delay in port activation after a node is powered up, and this may cause the DHCP process to fail because the switch is not ready. For ports on the cluster VLAN, where we have tight control of the configuration, we disabled the spanning tree protocol to prevent this problem.

The Cisco Discovery Protocol (CDP) is used by the Cisco switches to identify other switches in the network. Since the FastEthernet ports on the cluster and management LANs connect only to nodes, we have disabled CDP protocol on those ports.

*Example 6-2 Initial Cisco 3524XL configuration file*

---

```
!  
version 12.0  
no service pad  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname cisco001  
!  
enable secret 5 $1$HWuA$5FJfDhgM73Cy2U0ca5Zca.  
!  
!  
!  
!  
!  
ip subnet-zero  
!  
!  
interface FastEthernet0/1  
!  
interface FastEthernet0/2  
!  
interface FastEthernet0/3  
  
... [duplicate lines removed]...  
  
!  
interface FastEthernet0/22  
!  
interface FastEthernet0/23  
!
```

```

interface FastEthernet0/24
!
interface GigabitEthernet0/1
  switchport mode trunk
!
interface GigabitEthernet0/2
!
interface VLAN1
  ip address 10.1.1.241 255.255.0.0
  no ip directed-broadcast
  no ip route-cache
!
snmp-server engineID local 0000000902000008A3B32240
snmp-server community private RW
snmp-server community public RO
!
line con 0
  transport input none
  stopbits 1
line vty 0 4
  password cisco
  login
line vty 5 15
  password cisco
  login
!
end

```

---

*Example 6-3 Completed Cisco 3524XL configuration file*

---

```

!
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname cisco001
!
enable secret 5 $1$HWuA$5FJfDhgM73Cy2U0ca5Zca.
!
!
!
!
!
!
ip subnet-zero
!

```

```
!  
!  
interface FastEthernet0/1  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/2  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/3  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/4  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/5  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/6  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/7  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/8  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/9  
  switchport access vlan 2  
  spanning-tree portfast  
  no cdp enable  
!  
interface FastEthernet0/10  
  switchport access vlan 1
```



```
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/11
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/12
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/13
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/14
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/15
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/16
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/17
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/18
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/19
switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/20
```

```

switchport access vlan 1
spanning-tree portfast
no cdp enable
!
interface FastEthernet0/21
switchport access vlan 3
no cdp enable
!
interface FastEthernet0/22
switchport access vlan 3
no cdp enable
!
interface FastEthernet0/23
switchport access vlan 3
no cdp enable
!
interface FastEthernet0/24
switchport access vlan 3
no cdp enable
!
interface GigabitEthernet0/1
switchport mode trunk
!
interface GigabitEthernet0/2
switchport access vlan 2
spanning-tree portfast
no cdp enable
!
interface VLAN1
ip address 10.1.1.241 255.255.0.0
no ip directed-broadcast
no ip route-cache
!
snmp-server engineID local 0000000902000008A3B32240
snmp-server community private RW
snmp-server community public RO
!
line con 0
transport input none
stopbits 1
line vty 0 4
password cisco
login
line vty 5 15
password cisco
login
!
end

```

!

---

5. Transfer the edited configuration file back to the Cisco switch and activate it using the steps shown in Example 6-4.

*Example 6-4 Transferring the completed configuration file back to the switch*

---

```
cisco001#copy tftp running-config
Address or name of remote host []? 10.1.0.1
Source filename []? cisco001-config
Destination filename [running-config]?
Accessing tftp://10.1.0.1/cisco001-config...
Loading cisco001-config from 10.1.0.1 (via VLAN1): !
[OK - 2964 bytes]

2964 bytes copied in 2.926 secs (1482 bytes/sec)
cisco001#
00:24:13: %SYS-5-CONFIG: Configured from 10.1.0.1 by
cisco001#copy running-config startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
cisco001#
```

---

This completes the configuration of your network switch. Save the cisco001-config file as a record of your configuration.

## 6.1.2 Management Processor Adapter setup

Now that the master node and network are set up, we need to begin controlling the nodes. This is accomplished through the MPA cards and the onboard service processors on the nodes. In our lab we used the Remote Supervisor Adapter (RSA) card. xCAT also supports the earlier Advanced Systems Management Adapter (ASMA) card. Each card must be programmed with an IP address, a unique identification, and SNMP monitoring options.

### RSA card setup

To set up the RSA card:

1. Download xSeries 330 - Remote Supervisor Adapter Firmware Update (Version 1.03 at the time of this writing).
2. Create the RSA config floppy:

Using DOS (be certain that you use *command* and not *cmd* under NT or Win2k), run the .exe and follow the prompts.

Boot the node containing the RSA card using the configuration floppy. At the configuration prompt, enter a basic configuration. Select **Configuration Settings -> Systems Management Adapter** and apply the following changes for Ethernet settings:

- To use static IP addressing:
  - Enable network interface
  - Set local IP address for RSA network interface
  - Set subnet mask
  - Set DHCP setting to DISABLED
  - Press F6 to commit changes
- To use DHCP addressing instead:
  - Enable network interface
  - Set DHCP setting to ENABLED
  - Press F6 to commit changes

You must then record the RSA card's MAC address in the `mac.tab` file and refresh the DHCP configuration using `makedhcp rsa-hostname`. You can find the MAC address via the View Advanced Network Settings menu item.

3. Restart the RSA adapter using F9 from the Ethernet Settings menu.

Check that the management processors are on the network, using the command `pping mpa`. This does a parallel `ping` to all of the adapters you defined as being part of the `mpa` group. If there are problems communicating with a Management Adapter, insure that the host name is resolving to the correct IP address, that the network port on the Ethernet switch shows a connection, and that the adapter has been restarted after saving the configuration changes.

## Programming the Management Processor Adapter

The xCAT script `mpasetup` is used to set up the alert, SNMP traps, and other configuration settings onto the MPA cards. It may be run on a single MPA card or a range defined in `nodelist.tab`. See Example 6-5.

*Example 6-5 Output of the `mpasetup` command*

---

```
[root]# mpasetup rsa001
rsa001: SUCCESS: setdhcp -enabled type=boolean false
rsa001: SUCCESS: setip -hostname type=String rsa001
rsa001: SUCCESS: setip -ipaddress type=String 10.1.1.101
rsa001: SUCCESS: setip -subnet type=String 255.255.0.0
rsa001: SUCCESS: setsnmp -traps type=boolean true
rsa001: SUCCESS: setsnmp -communityname type=int type=String 1 public
rsa001: FAILURE: setsnmp -ipaddress type=int type=int type=String 1 1 noip
Invalid parameter.
rsa001: FAILURE: setsnmp -agent type=boolean true Command failed.
rsa001: SUCCESS: setmpid -text type=String rsa001
```

```

rsa001: SUCCESS: setmpid -numeric type=String 10001
rsa001: SUCCESS: setalertentry -enabled type=boolean true
rsa001: SUCCESS: setalertentry -type type=String snmp.lan
rsa001: SUCCESS: setalertentry -ipaddress type=String noip
rsa001: SUCCESS: setalertentry -emailaddress type=String root@noip
rsa001: SUCCESS: setalertentry -criticaleventsonly type=boolean true
rsa001: SUCCESS: setalerttrigger -enabled type=String noncritical.voltage
rsa001: SUCCESS: setalerttrigger -enabled type=String noncritical.temperature
rsa001: SUCCESS: setalerttrigger -enabled type=String noncritical.single_fan
rsa001: SUCCESS: setalerttrigger -enabled type=String noncritical.rps
rsa001: SUCCESS: setalerttrigger -enabled type=String
noncritical.expansion_device
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.vrm
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.voltage
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.temp
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.tamper
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.power_supply
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.multiple_fan
rsa001: SUCCESS: setalerttrigger -enabled type=String critical.dasd
rsa001: SUCCESS: setalerttrigger -enabled type=String system.post
rsa001: SUCCESS: setalerttrigger -enabled type=String system.os
rsa001: SUCCESS: setalerttrigger -enabled type=String system.loader
rsa001: SUCCESS: setalerttrigger -enabled type=String system.application
rsa001: SUCCESS: setalerttrigger -enabled type=String system.power_off
rsa001: SUCCESS: setalerttrigger -enabled type=String system.power_on
rsa001: SUCCESS: setalerttrigger -enabled type=String system.boot
rsa001: SUCCESS: setalerttrigger -enabled type=String system.pfa
rsa001: Restarting MP, please wait...
rsa001: SUCCESS: restartmp -flag default
rsa001: PASSED: The management processor has been successfully restarted
[root]# mpacheck mpa
rsa001: 10.1.1.101 255.255.0.0 auto half public 0.0.0.0
rsa002: 10.1.1.102 255.255.0.0 auto half public 0.0.0.0

```

---

Verify that the management processors were programmed correctly using the command **mpacheck**. This command confirms that communications can be established with the specified MPA card(s).

### 6.1.3 Terminal server setup

Each terminal server must be configured with an IP address and serial port settings appropriate to Linux console windows. Terminal servers commonly used on IBM @server xSeries clusters include the Equinox ELS-II and ESP-16 models. In our lab we used the ELS-II; see Appendix D, "Application examples" on page 225, for ESP-16 procedures.

## Equinox ELS-II

The ELS-II from Equinox is a powerful device that supports virtually any combination of serial access you can think of. The manual is large and comprehensive, but there is no need to be put off by the size of the manual; configuring, for our purposes (known as *reverse telnet*), is simple.

The configuration procedure comes in two stages, the first over a serial cable and the second over the Ethernet. If your cluster is over 16 nodes you will have multiple ELSs so you will need to perform this procedure multiple times.

## Assigning an IP

Before you configure the ELS it is a good idea to reset it to the factory defaults. Ensure that the ELS is powered up, then locate the reset hole to the left of the Ethernet connector. Use a paperclip or similar item to hold the switch down until the LEDs on the ELS start to flash. The reset switch requires only moderate pressure; do not press too hard or you will cause the switch to stick. When the LEDs stop flashing, the ELS has been reset to the factory defaults. For further information, please refer to the Equinox manual.

In order to assign an IP to an ELS you must connect a system (your management node, a laptop, or any other computer) to one of the Equinox serial ports. If your ELS is fully populated with cables you will need to temporarily unplug one from a node. To use your management node to set up the ELS, connect the DB-9 adaptor Equinox part #210062 to one of the management node's available serial ports and connect a serial cable from the ELS to the DB-9 adapter. You can test that the serial connection is good with:

```
> cu -l /dev/ttyS1 -s 9600
```

Press Enter to connect and you should see:

```
Username>
```

Unplug the serial cable to have `cu` hang-up and then reconnect it for the next step.

```
> setupeisip ELS_HOSTNAME ttyS1
```

Once you have disconnected, verify that the IP has been assigned correctly by pinging the ELS:

```
> ping ELS_HOSTNAME
```

In order to proceed with the next step of the configuration you should disconnect the serial cable from the configuration system. If you borrowed the cable from one of your nodes, do not forget to put it back again.

After assigning the ELS IP address over the serial link, use the following to finish the setup for each ELS in your cluster. This sets up the terminal server's serial settings. See Example 6-6.

```
> setupels ELS_HOSTNAME
```

**Note:** After the serial settings are set, you cannot use `setupelsip` again, because the serial ports have been set for reverse use. If you need to change the IP address or reset the port setup, reset the unit to defaults as previously described, and repeat the entire configuration process.

*Example 6-6 Output of setupels command*

---

```
[root]# /usr/local/xcat/sbin/setupels els001
Trying 10.1.1.161...
Connected to els001.
Escape character is '^'.
```

```
#>
```

```
LAT_00807D20FD0C::Remote Console
Equinox Systems Inc., ELS-16
Welcome to the Equinox Server Network
```

```
Local> set priv
Password>
Local>> define port 1-16 access remote
Local>> define port 1-16 flow control enabled
Local>> define port 1-16 speed 9600
Local>> change port 1-16 que disable
Local>> lo port 1-16
Local>>
[root]#
```

---

Verify that you can `telnet` to port 3001 of the ELS. You should get a connection, but since there is nothing active on the port you will see no activity. Close the telnet session using Ctrl and ]. See Example 6-7.

#### Example 6-7 Testing the telnet connection

---

```
[root]# telnet els001 3001
Trying 10.1.1.161...
Connected to els001.
Escape character is '^]'.
^]
(to exit the telnet command press: control and ] )
telnet> close
Connection closed.
[root]#
```

---

Your ELS is now set up. Go back and repeat the procedure for any additional ELSs you may have.

**Note:** Most terminal servers implement a *reverse telnet* protocol, and will be configured in a similar fashion. The Equinox ESP series uses a different process. Examples of how to set up an ESP, which uses a local port model, and an iTouch, which also uses *reverse telnet*, are given in Appendix C, "Other hardware components" on page 211.

### 6.1.4 APC MasterSwitch setup

If an APC MasterSwitch is used for remote power control of equipment that does not have onboard power control, we recommend configuring the APC with DHCP. Configure `nodelist.tab` and `mac.tab` manually, and run the `makedhcp apc-hostname` command.

**Tip:** If you want to use fix IP address, here is an example of how to configure APC with fix IP address:

1. Connect a serial terminal or terminal emulator to the APC's serial port at 2400 baud, 8 bits, no parity, 1 stop bit.
2. Enter the system ID and password (the default is `apc` for both).
3. Select option 2 (Network).
4. Select option 1 (TCP/IP).
5. Set the IP address, subnet mask, and gateway (use the IP address of the master node).
6. Accept the changes by using option 5.
7. Return to the main menu (press Esc twice) and log out (4).



Confirm you can now **ping** the switch by name and IP address from the master node. If you wish to do additional configuration, you may do it through the unit's Web interface, consult the manual that came with the APC switch, or download it from <http://www.apc.com/> (**Support -> User Manuals -> Power Distribution -> AC Power Controllers**).

**Tip:** If the availability of outlets requires you to plug the network switch into the APC MasterSwitch, you should consider configuring that port to Never Off, in order to avoid turning off the network switch and losing access to the APC MasterSwitch over the network. Should this happen, you can recover by using the serial terminal connection as previously described.

## 6.1.5 BIOS and firmware updates

For consistency, all cluster nodes of the same model should have the same version of the BIOS installed. These BIOS files can be downloaded from the IBM Support Web site <http://www.pc.ibm.com/support>. This site is shown in Figure 4-1 on page 44. Enter your model number and select **Downloadable Files** from the menu on the left-hand side. You will also find the latest firmware updates for all the other components in all your nodes, such as the ServeRAID adapter, the RSA card, and so on. You can download a complete System Service Package CD-ROM image containing all the images for your node type, or individual images for each firmware element (BIOS, onboard diagnostics, etc). Check to see if individual images have been updated since the System Service Package, and download those components separately. Follow the instructions provided with the images to create and run the updates. Note that you can modify the BIOS flash diskette to run in unattended mode by changing the CONFIG.SYS file to read:

```
SHELL=FLASH2.EXE /U
```

(The readme file on the diskette may provide incorrect instructions for this).

**Attention:** If it becomes necessary to replace the system board on one of your nodes, check the BIOS level of the replacement board before re-flashing. Although it is desirable to maintain a common BIOS level on all nodes, you should never attempt to load an older BIOS version than what is provided on a replacement board unless you are explicitly instructed to do so by IBM support.

After flashing your components you have to update the BIOS settings. BIOS configuration is accomplished using the BIOS setup program, which is started by pressing the F1 key on the keyboard when the node is powered on. To be sure you have consistent settings, start by resetting to default, then change the settings where a non-default value is needed. The settings we used (with the values emphasized) are:

► Under Devices and I/O Ports:

Serial Port A	Port 3F8, IRQ 4
Serial Port B	Disabled

– Under Remote Console Redirection:

Active	<i>Enabled</i>
Port	COM1
Baud Rate	9600
Data Bits	8
Parity	None
Stop Bits	1
Emulation	ANSI
Active After Boot	<i>Enabled</i>

► Under Start Options:

Boot Fail Count	<i>Disabled</i>
Virus Detection	Disabled

– Under Startup Sequence:

First Startup Device	<i>Diskette Drive 0</i>
Second Startup Device	<i>CD ROM</i>
Third Startup Device	<i>Network</i>
Fourth Startup Device	<i>Hard Disk 0</i>

**Note:** The example above is for the 8674 model of the x330. Other models may have slightly different prompt sequences.

**Tip:** A utility is provided on the BIOS flash diskette for saving and restoring the BIOS settings from CMOS memory. This utility can be used to transfer BIOS settings from one node to another provided that the nodes are the same model and have the same memory and PCI card configuration. You can create a utility diskette to do this by using these steps:

1. Create a copy of the BIOS update disk or download the BIOS update diskette image and create a boot diskette according to the instructions provided.
2. Edit the file CONFIG.SYS on the diskette to run the CMOS utility program to save configuration settings to a file (for example, X330CMOS.DAT):

```
SHELL=CMOSUTIL.EXE /S X330CMOS.DATdat
```

This save file must not already exist on the disk; you must delete or rename it if you want to redo this step.

3. Put this diskette in the node you configured manually and reboot. You should see a message indicating that the CMOS image file was created, followed by a command interpreter error, which you may ignore.
4. Edit the file CONFIG.SYS on the diskette to restore the CMOS settings using the same utility program:

```
SHELL=CMOSUTIL.EXE /R X330CMOS.DATdat
```

Booting a node from this diskette will now restore your saved CMOS settings. You may wish to make multiple copies of this diskette if you have many nodes to prepare, and prepare one for each unique combination of node model, memory size, and PCI card you have. It is helpful to keep a copy of this diskette with your cluster to use in case a system board must be replaced.

**Restriction:** At the time of this writing, the CMOSUTIL on the BIOS diskette (1.03) for newer model (8674) x330 machines does not restore the settings of the serial port I/O addresses, IRQ's, or any of the settings of the Remote Console Redirection. If you wish to use these options, you must run setup on each node manually. This problem will be corrected on the next BIOS diskette, and an unofficial fix is available from the <http://www.x-cat.org> Web site.

There is also a set of utilities to automate and simplify the firmware update procedure available from <http://www.x-cat.org>. In order to use these utilities, you need access to a CD-R writer and software capable of creating a CD-R from an iso image file. The utilities are provided in the form of two files, stage1.dd and stage1.iso. The .dd file is a diskette image, and the .iso file is a CD-ROM image. Make the stage1 diskette by using the command:

```
[root]# dd if=stage1.dd of=/dev/fd0 bs=512
```

Make the stage1 CD-ROM using your CD-R burner software and the stage1.iso file.

**Note:** The stage1.iso file contains firmware images downloaded from the IBM service Web site. There may be some lag between new firmware images being released on the IBM Support Web site and incorporating those images into the images provided on <http://www.x-cat.org>. Always check the IBM Support Web site for the most current updates.

Insert the stage1 diskette and the stage1 CD-ROM into a node and reboot it. This will update all of the firmware on the node to the versions on the CD-ROM.

Finally, the xCAT developers are working on implementing BIOS flashing over the network, as also known as xCAT Remote Flash. More information about this new feature can be found at:

<http://x-cat.org/docs/flash-H0WT0.html>

## 6.2 Stage 2: MAC address collection

Each Ethernet network interface is identified by a hardware address, the Media Access Control (MAC) address. This byte string is assigned by the manufacturer and is unique to that physical interface. The network boot process relies on the MAC address to identify a specific node before an IP address has been assigned to it. To insure each node is assigned the desired IP address, we must collect the MAC addresses from the nodes and store them in a table, which is used to generate the DHCP configuration file. Once this is done, each node will have a fixed IP address assigned after boot. During the boot process, each node will use DHCP to obtain an IP address. Then each node will query the master node as to whether it should network boot or load the OS from the local disk.

There are two methods used by xCAT to collect MAC addresses: Through the serial consoles and terminal servers, or through the Cisco Ethernet switch (if your cluster uses a Cisco switch). It is essential that the cables for the serial consoles (serial port method) or primary Ethernet (Cisco switch method) connections are plugged into known ports, and that the xCAT `cisco3500xl.tab` or `conserver.tab` tables correctly reflect the physical cabling. The Ethernet switch method only works with Cisco switches but, if you have a Cisco switch, this method allows you to proceed with your cluster configuration without first debugging the terminal server configuration. We recommend that you wire both the Ethernet ports and the terminal server ports sequentially to avoid confusion and allow you to use either method. Other than the means of obtaining the MAC addresses, the two methods are essentially the same. In our lab we used the Cisco switch method of MAC address collection.

At this point it is generally best to test the stage 2 procedure on a single node to confirm that everything is working correctly. Once you have one node done, the rest can be done much more quickly.

1. Manually reset the nodes for which you want to collect MAC address data, via the power or reset buttons. As the nodes do not get a unique personality assigned until after stage 3, the remote power controls provided by the MPN are not yet available.
2. Each node will PXE boot at a temporary, dynamic IP address from the range defined by the keyword `dynamicip` in `site.tab`, load a Linux kernel, and run a small script that loops endlessly echoing the output shown in Example 6-8. The MAC address can then be captured on the serial port using the terminal servers, or from the MAC address table on the Cisco switch.

*Example 6-8 Stage 2 node MAC loop*

---

```
--- 10.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
MAC-00:02:55:C6:5A:0D-MAC
MAC-00:02:55:C6:5A:0D-MAC
MAC-00:02:55:C6:5A:0D-MAC
PING 10.0.0.1 (10.0.0.1) from 10.9.1.1 : 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=0 ttl=255 time=0.1 ms

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
MAC-00:02:55:C6:5A:0D-MAC
MAC-00:02:55:C6:5A:0D-MAC
MAC-00:02:55:C6:5A:0D-MAC
PING 10.0.0.1 (10.0.0.1) from 10.9.1.1 : 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=0 ttl=255 time=0.1 ms

--- 10.0.0.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
```

---

3. After giving all of the nodes time to boot up to this phase, enter the following command to retrieve the MAC information from the Cisco switch or the serial console port. The expected output of `getmacs` is shown in Example 6-9.

```
[root]# getmacs compute
```

*Example 6-9 Output of getmacs command*

---

```
[root]# getmacs node001-node008
```

```
Please reset nodes: node001 node002 node003 node004 node005 node006 node007
```

node008

Press [Enter] when ready...

Saving output to mac.lst in current directory /opt/xcat/etc.

```
node001 00:02:55:c6:5a:0d
node002 00:02:55:c6:59:5b
node003 00:02:55:c6:5a:05
node004 00:02:55:c6:59:fd
node005 00:02:55:c6:5a:17
node006 00:02:55:c6:62:ef
node007 00:02:55:c6:07:d6
node008 00:02:55:c6:5a:8e
```

Auto merge mac.lst with /opt/xcat/etc/mac.tab(y/n)? y  
[root]#

---

Reply y to accept the captured MAC addresses.

4. Merge these MAC addresses into your DHCP configuration with the command:

```
[root]# makedhcp compute
```

**Tip:** Some things to look for if you have trouble getting stage 2 to run correctly:

- ▶ Is your DHCP server running? The node must acquire an IP address to boot.
- ▶ Is your TFTP server running? PXELinux uses TFTP to load the kernel and initrd.
- ▶ Are your nodes set to network boot?

If using a Cisco switch to acquire MAC addresses:

- ▶ Does the password in `passwd.tab` match what you configured in the switch?

If using terminal servers to acquire MAC addresses:

- ▶ Is your serial terminal subsystem working? If you have enabled Remote Console Redirection in the BIOS, you should see the same Power-On Self Test sequences in a `wcons` window that you see on the VGA screen for the node.
- ▶ Is conserver running? Try restarting it using the command `service conserver restart`.
- ▶ Are the terminal servers correctly connected? Do you have the correct RJ45 to DB-9 adapters? These are *not* interchangeable between terminal server models.

## 6.3 Stage 3: Management processor setup

We will now program the onboard service processor on each node. This is similar to stage 2 in that a small kernel is loaded on each node to perform a specific function. In this case, a shell script is run, which loads the Linux Service Processor driver and command line utility and uses inband commands to program the service processor. This procedure sets alerts and a unique name for each service processor just as was previously done for the management adapters. These names are taken from `mp.tab`.

It may be helpful to monitor the nodes being programmed via their serial consoles using a command like `wcons -t 8 rack01` according to the group of nodes being programmed. You should also watch the system log by opening a window on the management node and entering the command:

```
tail -f /var/log/messages
```

1. Put the nodes to be programmed into stage 3 mode with the command:

```
nodeset compute stage3
```

2. Reboot the nodes manually using the power or reset buttons. You should then see the MPN procedure move forward smoothly in all the compute nodes' `rvitals` windows, as shown in Example 6-10.

*Example 6-10 MPN setup output from an x330 node*

---

```
setallalertsfor SNMP:
Enable Alerts Entry : Result = success
Setting Alert Text Name : Result = success
Setting SNMP Alerts : Result = success
Setting Temp Alerts : Result = success
Setting Voltage Alerts : Result = success
Setting Tamper Alerts : Result = success
Setting Multiple Fan Alerts : Result = success
Setting Power Alerts : Result = success
Setting HD Alerts : Result = success
Setting VRM Alerts : Result = success
Setting Redundant Power Alerts : Result = success
Setting One Fan Alerts : Result = success
Setting Non Crit Temp Alerts : Result = success
Setting Non Crit Voltage Alerts : Result = success
Setting 2nd Device Alerts : Result = unsupported
Setting POST Hang Alerts : Result = success
Setting OS Hang Alerts : Result = success
Setting App Logged Error Alerts : Result = success
Setting System Power Off Alerts : Result = success
Setting System Power On Alerts : Result = success
Setting System Boot Failure Alerts : Result = success
Setting Loader Watchdog Failure Alerts : Result = success
Setting PFA Alerts : Result = success

noerrorlogfullevents:
Disable Error Log Full Event : Result = success

setassettag:
Setting SP Asset Tag = 00:02:55:C6:5A:0D : Result = success

setcom1:
Setting COM1 baud rate = 9600 : Result = success

setcom2:
Setting COM2 baud rate = 9600 : Result = success

setdialin:
Setting Login#1 ID = USERID : Result = success
Setting Login#1 password = PASSWORD : Result = success
```

---



3. To verify the results of the stage 3 process, use the command **mpascan** to see the names of the nodes attached to each mpa card. The **mpncheck** command can automatically check the management processor network (Example 6-11).

*Example 6-11 Output of the mpascan command after stage 3*

---

```
[root]# mpascan rsa001
rsa001: node001 3
rsa001: node002 0
rsa001: node003 1
rsa001: node004 5
rsa001: node005 2
rsa001: node007 4
rsa001: node008 6
[root]# mpncheck mpn1
node001: rsa001
node002: rsa001
node003: rsa001
node004: rsa001
node005: rsa001
node006: rsa001
node007: rsa001
node008: rsa001
```

---

Your MPN is now configured, and you have full access to the hardware controls and can perform power and monitoring with **rpower**, **rvitals**, etc. A few examples are shown in Example 6-12.

*Example 6-12 MPN functions*

---

```
[root]# rpower compute state
node001: on
node002: on
node003: on
node004: off
node005: on
node006: on
node007: on
node008: off
[root]# rpower node004,node008 on
node004: on
node008: on
[root]# rvitals node002 all
node002: CPU 1 Temperature = 38.0 C (100.4 F)
node002: CPU 2 Temperature = 40.0 C (104.0 F)
node002: hard shutdown: 95.0 C (203.0 F)
node002: soft shutdown: 90.0 C (194.0 F)
node002: warning: 85.0 C (185.0 F)
node002: warning reset: 78.0 C (172.4 F)
```

```
node002: DASD 1 Temperature = 31.0 C (87.8 F)
node002: Ambient Temperature = 25.0 C (77.0 F)
node002: System Board 5V: 5.04
node002: System Board 3V: 3.29
node002: System Board 12V: 12.17
node002: System Board 2.5V: 2.65
node002: VRM1: 1.41
node002: VRM2: 1.41
node002: Fan 1: 78%
node002: Fan 2: 79%
node002: Fan 3: 82%
node002: Fan 4: 79%
node002: Fan 5: 79%
node002: Fan 6: 78%
node002: Power is on.
node002: System uptime = 2924
node002: The number of system restarts = 105
node002: System State: OS running.
[root]# rinv node005 all
node005: CPU Speed = 1133
node005: Maximum DIMMs = 4
node005: Installed DIMMs: 1 2 3 4
node005: Memory Size = 1024
node005: machine model: 867411X
node005: serial number: 10AC1RP
node005: Asset Tag: 00:02:55:C6:5A:17
node005: VPD BIOS EME110AUS 1.0.3 4/23/2002
node005: VPD ASMP TTET27A 1.0.4 2/1/2002
[root]#
```

---

**Note:** Any time the motherboard must be changed on a node, the MAC address will change and the MPN configuration will be lost. You can rerun stage 2 and stage 3 on the node without affecting the Linux image installed on the system disk.

**Tip:** If you have problems with stage 3, here are some things to check:

- ▶ Check network booting tips for stage 2.
- ▶ To reset the integrated service processor on a node, you must remove power from the node. The power led must be *off*, not blinking.
- ▶ Check for VPD errors when you reboot a node.
- ▶ If you cannot connect to the service processor on a node, check it with the configuration utility on the Service Processor Firmware Update diskette for your system model. You can also use this utility to reflash the service processor firmware if it has become corrupted.

## 6.4 Stage 4: Node installation

The next step of the installation process is to adjust your Kickstart templates to your specific configuration and preferences. We suggest you save the original template provided in xCAT before editing it. The name will be determined by the value you have set in `nodetype.tab` for this node. You will find the Kickstart template in the directory `/opt/xcat/ksXX` where `XX` is the major Red Hat release being used. This template is in the format of a standard Red Hat Kickstart file, with additional substitution being done automatically by the xCAT scripts for variables designated with the `#` character. For example, the `#NFS_INSTALL#` variable is replaced with the name of the master node or staging node from which a given node is to obtain its installation images from. In this file you specify the items you would normally type during a manual Red Hat installation—the language, disk partitioning information, firewall configuration, time zone, the list of packages to be installed, etc. This is where you specify how you want your disk partitions to be set up, which portions of the Red Hat distribution you want to install on the compute nodes, and which kernel version you wish to run.

### 6.4.1 Creating a template file

xCAT's provided `.kstmp` file is a good start, but you will (at a minimum) want to insure that the disk partitioning and kernel version are appropriate for you. Set the `KERNELVER` parameter to the name of the kernel RPM package you wish to install. There is also an extensive post-install section in which the node is customized. xCAT's sample `.kstmp` files provide for substantial automatic customization, and you can add additional scripting to the `.kstmp` file to automate those customizations specific to your site. When making global changes to your nodes we recommend scripting those changes into the `.kstmp` files and reinstalling your nodes. While this may seem more time consuming than just using parallel shell commands to implement a change on all nodes, it results in a `.kstmp` file that accurately reflects the configuration of your nodes so they may

easily be reinstalled, and insures that all nodes get identical configurations. The post-install section of the .kstmpt file is standard bash script, and you can add any specific node customization you need for your installation.

The xCAT post-installation script already provides the following functions. (This list assumes you are using Red Hat 7.3; directories exist for other distributions.)

- ▶ Sets up remote logging so that the node logs are all sent to the master node.
- ▶ Sets up networking, including IP addresses, name resolution, and remote shells (**rsh** and/or **ssh**).
- ▶ Installs Red Hat updates from the directory `/install/post/updates/rh73` on the master node.
- ▶ Installs any optional RPMs you have placed in `/install/post/rpm73`.
- ▶ Installs any tarballs you have placed in `/install/post/tarballs`.
- ▶ Installs an updated Linux kernel. This can be either a Red Hat kernel in RPM format, placed in `/install/post/updates/rh73`, or a custom kernel in tarball format placed in `/install/post/kernel`.

In our lab we changed the `KERNELVER` entry in `compute73.kstmpt` to reflect our updated kernel's version.

```
[root]# vi /opt/xcat/ks73/compute73.kstmpt
KERNELVER=2.4.18-4smp
```

This:

- ▶ Configures the boot loader (LILO) to access the updated kernel. xCAT only supports the LILO bootloader at this time; do not select GRUB or xCAT will not be able to activate your updated kernel.
- ▶ Sets up NFS mounts for the `/usr/local` and `/home` file systems on the master node.
- ▶ Synchronizes the node's clock and sets up `ntp` to keep it in sync with the master node.
- ▶ Sets up PAM for access security.
- ▶ Sets up `/etc/rc.local` to adjust kernel tuning.
- ▶ Deletes unneeded services.
- ▶ Copies modified system files from `/install/post/sync`. Subdirectories may be set up in this directory by node type, node resource (from `noderes.tab`), or individual host names. The original file will be renamed with a `.ORIG` suffix, and the files from this directory copied in. For example, the file `/install/post/sync/compute73/etc/hosts.allow` would be installed as `/etc/hosts.allow` on any node of the `compute73` type.

- ▶ Sets up serial consoles to be used by the terminal servers.

After completing your `.kstmp` file, it must be processed to resolve all of the variables and to produce unique files for each possible permutation of the file across multiple node types and staging nodes. The `mkks` script also validates a number of the xCAT tables to insure that the required information has been provided.

```
cd /opt/xcat/ks73
./mkks
```

Correct any errors reported by `mkks`, and repeat the command.

## 6.4.2 Creating a custom kernel RPM image

Linux offers a great deal of flexibility in the configuration of the kernel, and some cluster users find that the standard Red Hat kernels are not suitably configured, or that they prefer to run the latest release of the kernel from the official Linux kernel Web site <http://www.kernel.org>. xCAT provides for this flexibility by allowing you to create your own kernel image. After you have configured your kernel and compiled and tested it, use the command `make rpm` from the top directory of your kernel source tree to create the RPM file in the directory `/usr/src/redhat/RPMS/i386`. You can then copy this RPM into the `/install/post/updates/kernel` directory and have it installed just as you would a Red Hat kernel. Be sure to set `EXTRAVERSION` in the `.config` file for your kernel to identify it, and set `KERNELVER` in your `.kstmp` file to include your `EXTRAVERSION`.

## 6.4.3 Creating a custom kernel tarball image

xCat also supports saving your custom kernel in a tarball format. To create a kernel tarball, configure and compile your kernel, and then create an absolute tar file containing the kernel, the modules, the `initrd` (if one is used), and optionally the kernel source tree. For this example, we set `EXTRAVERSION` in the kernel makefile to `ibmsmp`.

```
[root]# tar -czvf /install/post/kernel/kernel-2.4.18-ibmsmp.tgz -C /
boot/System-map-2.4.18-ibmsmp boot/System-map boot/initrd-2.4.18-ibmsmp.img
boot/vmlinuz-2.4.18-ibmsmp lib/modules/2.4.18-ibmsmp
```

By placing this kernel tarball image in `/install/post/kernel` and setting the `KERNELVER` variable to `2.4.18-ibmsmp` in the `.kstmp` file, the post-install process will install and activate this kernel.

**Tip:** There is a script in `/opt/xcat/build/kernel/makekerneltgz` that creates a kernel tarball for you; it also works with Itanium-based servers.

## 6.4.4 Installing the nodes

When you have your `.kstmp` file completed, you are ready to begin installation. Once again, it is helpful to start by trying one node.

```
[root]# winstall node001
```

Watch the first install carefully. Once the remote syslog has been initialized in the post-install section, you can monitor `/var/log/messages` on the master node to watch the progress of the post-install scripts. (If you add your own script commands to the post-install section, use the `logger` command to show the progress of the installation in the system log, as in Example 6-13 on page 111.) After you have successfully installed your first node and checked that it is configured as you wish it to be, you are ready to install the rest of your cluster. The `winstall` command accepts a range of nodes, and will open tiny xterms on the master node. You can also use the `-t` and `-f` options to get the windows auto-arranged. Those windows can help you to observe the installation process. To zoom in on a specific node, use `Ctrl+right-click` and change the font from unreadable to medium or large.

To install all of our nodes, we used the command `winstall compute`. After resizing and rearranging some windows for clarity, we saw the installation screens shown in Figure 6-1 on page 111.

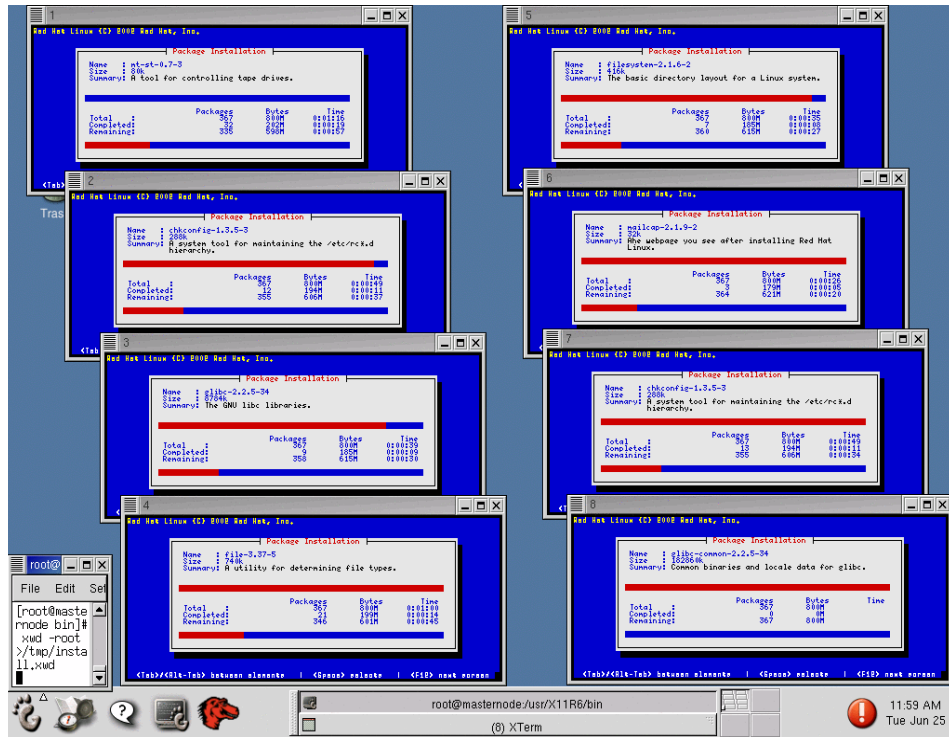


Figure 6-1 Installation screens

*Example 6-13 syslog entries for a single node install*

```
[root]# tail -f /var/log/messages
Jun 25 14:10:17 node001 exiting on signal 15
Jun 25 14:16:32 masternode dhcpd: DHCPDISCOVER from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:32 masternode dhcpd: DHCP OFFER on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:33 masternode dhcpd: DHCPREQUEST for 10.0.1.1 from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:33 masternode dhcpd: DHCPACK on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:34 masternode tftpd[16898]: Serving /tftpboot/pxelinux.0 to 10.0.1.1:2070
Jun 25 14:16:34 masternode tftpd[16899]: Serving /tftpboot/pxelinux.0 to 10.0.1.1:2071
Jun 25 14:16:35 masternode tftpd[16900]: Serving /tftpboot/pxelinux.cfg/0A000101 to 10.0.1.1:57217
Jun 25 14:16:35 masternode tftpd[16901]: Serving /tftpboot/xcat/ks73z to 10.0.1.1:57218
Jun 25 14:16:35 masternode tftpd[16902]: Serving /tftpboot/xcat/ks73.gz to 10.0.1.1:57219
Jun 25 14:16:53 masternode dhcpd: DHCPDISCOVER from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCP OFFER on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPDISCOVER from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCP OFFER on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPREQUEST for 10.0.1.1 from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPACK on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
```

```

Jun 25 14:16:53 masternode rpc.mountd: authenticated mount request from node001:612 for
/install/ks73 (/install)
Jun 25 14:16:53 masternode dhcpd: DHCPDISCOVER from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPOFFER on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPDISCOVER from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPOFFER on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPREQUEST for 10.0.1.1 from 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode dhcpd: DHCPACK on 10.0.1.1 to 00:02:55:c6:5a:0d via eth2
Jun 25 14:16:53 masternode rpc.mountd: authenticated mount request from node001:616 for
/install/rh73 (/install)
Jun 25 14:23:03 node001 syslogd 1.4.1: restart.
Jun 25 14:23:03 node001 syslog: syslogd startup succeeded
Jun 25 14:23:03 masternode rpc.mountd: authenticated mount request from node001:917 for
/install/post (/install)
Jun 25 14:23:03 node001 kernel: klogd 1.4.1, log source = /proc/kmsg started.
Jun 25 14:23:03 node001 syslog: klogd startup succeeded
Jun 25 14:23:03 node001 logger: Install: syslog setup
Jun 25 14:23:03 node001 logger: Install: mounting /post
Jun 25 14:23:03 node001 logger: Install: setting up eth0
Jun 25 14:23:03 node001 logger: Install: setting up eth1
Jun 25 14:23:03 node001 logger: Install: Update RH7.3 RPMS
Jun 25 14:23:06 node001 logger: Preparing...
#####
Jun 25 14:23:06 node001 logger: bind-utils
#####
Jun 25 14:23:09 node001 logger: dateconfig
#####
Jun 25 14:23:14 node001 logger: ghostscript
#####
Jun 25 14:23:19 node001 logger: LPRng
#####
Jun 25 14:23:21 node001 logger: nss_ldap
#####
Jun 25 14:23:22 node001 logger: perl-Digest-MD5
#####
Jun 25 14:23:24 node001 logger: Install: installing rpms
Jun 25 14:23:24 node001 logger: Install: installing rpm atftp-0.6-1.i386.rpm
Jun 25 14:23:24 node001 logger: Preparing...
#####
Jun 25 14:23:24 node001 logger: atftp
#####
Jun 25 14:23:25 node001 logger: Install: installing new RPM kernel
Jun 25 14:23:26 node001 logger: Preparing...
#####
Jun 25 14:23:29 node001 logger: kernel-smp
#####
Jun 25 14:23:38 node001 logger: Added linux
Jun 25 14:23:38 node001 logger: Added linux-up
Jun 25 14:23:38 node001 logger: Added xCAT *

```



```
Jun 25 14:23:38 node001 logger: Install: /post/kernel/gm-1.5.1_Linux-2.4.18-4smp.i686.rpm not
found! Setting up Myrinet anyway.
Jun 25 14:23:38 node001 logger: Install: setup /etc/ssh/sshd_config
Jun 25 14:23:38 node001 logger: Install: setup root .ssh
Jun 25 14:23:38 node001 logger: Install: setup scratch
Jun 25 14:23:38 node001 logger: Install: setup NFS mounts in fstab
Jun 25 14:23:38 node001 logger: Install: sync clock
Jun 25 14:23:38 node001 ntpdate[1961]: step time server 10.0.0.1 offset 2.544255 sec
Jun 25 14:23:40 node001 logger: Install: Setup NTP
Jun 25 14:23:40 node001 logger: Install: Setup rc.local
Jun 25 14:23:40 node001 logger: Install: Setup sendmail
Jun 25 14:23:40 node001 logger: Install: Setup man paths
Jun 25 14:23:40 node001 logger: Install: Setup PATHS
Jun 25 14:23:40 node001 logger: Install: Setup ls.so paths
Jun 25 14:23:40 node001 logger: Install: Setup services
Jun 25 14:23:40 node001 logger: Install: copying /post/sync/rh73 to /
Jun 25 14:23:40 masternode rpc.mountd: authenticated unmount request from node001:924 for
/install/post (/install)
Jun 25 14:23:40 node001 logger: /post/sync/rh73
Jun 25 14:23:40 node001 logger: 0 blocks
Jun 25 14:23:40 node001 logger: Install: unmounting /post
Jun 25 14:23:40 node001 logger: Install: setup serial console
Jun 25 14:23:40 node001 logger: LILO version 21.4-4, Copyright (C) 1992-1998 Werner Almesberger
Jun 25 14:23:40 node001 logger: 'lba32' extensions Copyright (C) 1999,2000 John Coffman
Jun 25 14:23:40 node001 logger:
Jun 25 14:23:40 node001 logger: Reading boot sector from /dev/sda
Jun 25 14:23:40 node001 logger: Merging with /boot/boot.b
Jun 25 14:23:40 node001 logger: Boot image: /boot/vmlinuz-2.4.18-3smp
Jun 25 14:23:40 masternode sshd(pam_unix)[16934]: session opened for user root by (uid=0)
Jun 25 14:23:40 node001 logger: Mapping RAM disk /boot/initrd-2.4.18-3smp.img
Jun 25 14:23:40 node001 logger: Added linux
Jun 25 14:23:40 node001 logger: Boot image: /boot/vmlinuz-2.4.18-3
Jun 25 14:23:40 node001 logger: Mapping RAM disk /boot/initrd-2.4.18-3.img
Jun 25 14:23:40 node001 logger: Added linux-up
Jun 25 14:23:40 node001 logger: Boot image: /boot/vmlinuz-2.4.18-4smp
Jun 25 14:23:40 node001 logger: Mapping RAM disk /boot/initrd-2.4.18-4smp.img
Jun 25 14:23:40 node001 logger: Added xCAT *
Jun 25 14:23:40 node001 logger: /boot/boot.0800 exists - no backup copy made.
Jun 25 14:23:41 node001 logger: Writing boot sector.
Jun 25 14:23:41 masternode sshd(pam_unix)[16934]: session closed for user root
Jun 25 14:23:41 node001 logger: Install: syslog setup again
Jun 25 14:23:41 node001 logger: Install: update local and remote installation flags
Jun 25 14:23:41 node001 logger: Warning: Permanently added 'masternode' (RSA1) to the list of
known hosts.^M
Jun 25 14:23:41 node001 logger: node001: boot
Jun 25 14:23:41 node001 exiting on signal 15
```

---

## 6.4.5 Post-installation

Once the installations have completed, your nodes will reboot using the newly installed Linux operating system. Most of the steps that need to be completed after Linux is installed on each of your nodes are done in the %post section of the .kstmp file. However, there is one remaining step that you must do manually to set up SSH keys. Like most xCAT commands this command can be applied to a node or range of nodes.

```
[root]# makesshgkh compute
Scanning keys, please wait...
```

You should now be able to use **ssh** to issue commands on individual nodes, and to issue parallel commands across groups of nodes. Example 6-14 shows the use of the **ssh** and **psh** (parallel shell) commands to insure that the correct kernel is installed on each node. Your basic cluster configuration is complete, and you are ready to install applications and start doing useful work. For ideas on some applications you might start with, see Appendix D, "Application examples" on page 225.

*Example 6-14 Using ssh and psh commands to access the nodes*

---

```
[root]# ssh node001 uname -rv
node001:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
[root]# psh compute uname -rv
node005:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node001:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node002:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node007:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node008:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node006:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node004:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
node003:      2.4.18-4smp #1 SMP Thu May 18 18:32:33 EDT 2002
```

---

**Tip:** If you have problems with network installs, here are some things to check.

If the base Red Hat install is not working:

- ▶ Check network booting tips for stage 2.
- ▶ Is NFS enabled on the master node?
- ▶ Is the `/install` file system NFS exported? Are the export permissions right?
- ▶ Is DNS on the master node up and working correctly? As a test, try changing a node's PXELinux config file in `/tftpboot/pxelinux.cfg/{Node IP in hex}` to fetch the Kickstart file from an IP address instead of a name. If this fixes your problem you have DNS problems that need to be resolved.
- ▶ Does the list of RPMs you have requested satisfy all dependencies?

If you are encountering errors in the post-install phase:

- ▶ Check the master node syslog for error messages. Be sure files you specify in the `.kstmp` script exist and can be accessed by the node during the installation phase. Keep in mind that the directory tree `/install/post` is mounted as `/post` on the node during customizations, and that other file systems are not automatically made available to you.
- ▶ Be sure you have satisfied the dependencies for any RPM packages you are trying to install.
- ▶ If you have added items to the `%post` section, use the `logger` command to track the progress of your additions. You may also pipe the output of commands into the `logger` if you need to debug your changes.





# A

## xCAT commands

This appendix provides a quick reference of the xCAT commands. Some of these commands are used only by xCAT scripts and should not be used by a non-experienced person. Other commands are for systems management and are normally used daily to administer the cluster environment.

We recommend that you take a look in the xCAT man pages before issuing a xCAT command. You can access the xCAT man pages at:

<http://x-cat.org/docs/man-pages/>

## Command reference

Table A-1 provides a quick reference of the xCAT commands. Next in this appendix, we provide a full description of each xCAT command, including the syntax, and examples on how to use them.

Table A-1 xCAT commands

Command	Description
<b>addclusteruser</b>	Adds a cluster user.
<b>mpacheck</b>	Checks MPA and MPA settings.
<b>mpaset</b>	Resets MPAs.
<b>mpascan</b>	Scans MPA for RS485 chained nodes.
<b>mpasetup</b>	Sets MPA settings.
<b>node1s</b>	Lists node characteristics from nodelist.tab, nodetype.tab, nodehm.tab, and noderes.tab.
<b>noderange</b>	Generates a list of node names.
<b>nodeset</b>	Sets the next cold or warm boot state for a list of nodes or groups.
<b>pping</b>	Pings a list of nodes in parallel.
<b>prcp</b>	Parallels remote copy.
<b>prsync</b>	Parallels rsync.
<b>psh</b>	Runs a command across a list of nodes in parallel.
<b>rcons</b>	Remote (text) console.
<b>revent</b>	Displays any number of remote hardware event log entries or clears them for a range of nodes.
<b>rinstall</b>	Forces an unattended network install for a range of nodes.
<b>rinv</b>	Retrieves hardware configuration information from the onboard management processor for a range of nodes.

Command	Description
<b>rpower</b>	Controls the power (on/off/stat) for a range of nodes.
<b>rreset</b>	Sends a hard reset (like pushing the reset button) to a range of nodes.
<b>rvid</b>	Redirects console text video. Useful for debugging boot issues.
<b>rvitals</b>	Retrieves hardware vital information from the onboard management processor for a range of nodes.
<b>wcons</b>	Remote (video) console.
<b>winstall</b>	Forces an unattended network install for a range of nodes. It is functionally equivalent to <b>rinstall</b> except that it will launch an xterm window for the remote console display.
<b>wvid</b>	Redirects console text video. It is functionally equivalent to <b>rvid</b> except that <b>wvid</b> will launch an xterm.

# addclusteruser - Add a cluster user

## Synopsis

addclusteruser [-h | --help | -v | --version]

## Description

The `addclusteruser` command interactively adds a single user. It will first determine if the `site.tab` fields, shown in Table A-2, are correctly set up.

Table A-2 *Site.tab* fields for `addclusteruser`

Field	Remarks
usermaster	Single node that is allowed to run <code>addclusteruser</code> . This is to prevent problems with environments that sync password and group files. This node is usually a user node or the NIS master.
nisdomain	NIS domain the cluster belongs to. NA if not using NIS.
nismaster	NIS master server for the cluster. NA if not using NIS.
chagemin	Minimum number of days between password changes.
chagemax	Maximum number of days before a password expires.
chagewarn	The number of days of warning before a password change is required.
chageinactive	After a password has expired, this is the number of days of inactivity before the account is locked.
homelinks	Defines a single directory for symbolic links to user directories. For example, a very large user community may have home directories on different mount points to manage load and growth. Many administrators use a <code>symlink</code> to the users' home directory so that if the directory is migrated in the future, only the link needs to be changed to maintain the user's environment. NA if not used.



After `site.tab` field validation, **addclusteruser** will interactively prompt for the user information and will not continue on error. The input prompts are presented in Table A-3.

Table A-3 *addclusteruser* prompts

Inputs	Description
username	User name.
group	User group. <b>addclusteruser</b> does not dynamically add user groups; groups must exist.
UID	User ID number. <b>addclusteruser</b> will assign the first available number starting with 500 if a UID is not specified.
home directory root	Root directory for home directories. For example, /home.
passwd	User password or blank for a random password.

After all the input is validated, **addclusteruser** creates the home directory and link (if specified by `homelinks`), then calls `useradd(8)`, `passwd(1)`, and `chage(1)` to create the user environment. **addclusteruser** then calls **gensshkeys(8)** and **genrhosts(8)** to set up the user's cluster node-to-node authentication environment. If NIS is set up, NIS will also be updated.

## Options

**-h | --help**            Print help.  
**-v | --version**        Print version.

## Files

**\$XCATROOT/etc/site.tab**      xCAT site file. See `site.tab(5)` for further details.

## Diagnostics

Errors should be self-explanatory.

## Examples

```
[root]# addclusteruser
Enter username: ibm
```

```
Enter group: users
Enter UID (return for next): 504
Enter absolute home directory root: /home
Enter passwd (blank for random): wMFISo37
Changing password for user ibm
passwd: all authentication tokens updated successfully
gmake[1]: Entering directory `~/var/yp/sensenet'
Updating passwd.byname...
Updating passwd.byuid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
gmake[1]: Leaving directory `~/var/yp/sensenet'
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

# mpacheck - Check MPA and MPA settings

## Synopsis

```
mpacheck [-l|-long|--long] noderange  
mpacheck [-h|--help|-v|--version]
```

## Description

**mpacheck** validates that xCAT can communicate with a range of IBM MPAs. MPAs must be listed as nodes in `nodelist.tab(5)`, `mpa.tab(5)`, and in `/etc/hosts` and/or DNS. **mpacheck** also returns the IP configuration of the MPA.

## Options

<b>-l   -long   --long</b>	Detailed listing.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b>\$XCATROOT/etc/mpa.tab</b>	xCAT management processor file. See <code>mpa.tab(5)</code> for further details.
<b>/etc/hosts</b>	System host table.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
MPA(s) are not defined in `nodelist.tab(5)`.
- ▶ `asmauser` not defined in `passwd.tab`  
Default user name is not defined in `passwd.tab(5)`.
- ▶ `asmapass` not defined in `passwd.tab`  
Default MPA password is not defined in `passwd.tab(5)`.
- ▶ Method not found  
MPA method defined by `nodehm.tab(5)` not found in `$XCATPREFIX/lib`.

- ▶ MPA type not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not ping MPA \$MPA  
\$MPA is the MPA defined in nodelist.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ mpcliroot not defined in site.tab  
See site.tab(5).
- ▶ MPA command not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not telnet to MPA \$MPA, and other telnet errors  
\$MPA is the MPA defined in nodelist.tab(5). Returned if telnet port check fails. PCI MPA may be in use or telnet interface is locked up. `mpareset(1)` can be used to reboot the PCI MPA or `rreset(1)` if MPA is powered by an APC MasterSwitch.

## Examples

```
[root]# mpacheck asma1
asma1: 192.168.1.11 255.255.255.0 100M Half public
192.168.1.1
Field 1: The MPA
Field 2: MPA IP
Field 3: MPA Subnet Mask
Field 4: Ethernet Speed (10/100/Auto)
Field 5: Duplex (Half/Full/Auto)
Field 6: SNMP Community 1
Field 7: SNMP IP Address 1
[root]# mpacheck --long asma1
asma1: Host IP Address 199.088.179.220
asma1: Host Subnet Mask 255.255.255.000
asma1: Gateway IP Address 199.088.179.022
asma1: Data Rate 4 - Ethernet 100M
asma1: Duplex 2 - Half Duplex
asma1: SNMP Traps Disable Off
asma1: SNMP Enable On
asma1: SNMP Community Name public
asma1: SNMP IP Address 1 199.088.179.022
```

## Bugs

Unknown.

## **Author**

Egan Ford (egan@us.ibm.com).

## **See also**

`noderange(3)`, `mpareset(1)`, `mpasetup(1)`, `mpascan(1)`, `mpncheck(1)`, `rreset(1)`

# mpareset - Reset MPAs

## Synopsis

```
mpareset noderange  
mpareset [-h|--help|-v|--version]
```

## Description

**mpareset** uses the SLIM or HTTP protocol to reboot a single or range of MPAs. MPAs must be listed as nodes in `nodelist.tab(5)` and in `/etc/hosts` and/or DNS. The reset method must also be defined in `mpa.tab(5)`.

## Options

<b>noderange</b>	See <code>noderange(3)</code> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b><code>\$XCATROOT/etc/nodelist.tab</code></b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/mpa.tab</code></b>	xCAT management processor file. See <code>mpa.tab(5)</code> for further details.
<b><code>/etc/hosts</code></b>	System host table.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
MPA(s) are not defined in `nodelist.tab(5)`.
- ▶ `asmauser` not defined in `passwd.tab`  
Default user name is not defined in `passwd.tab(5)`.
- ▶ `asmapass` not defined in `passwd.tab`  
Default MPA password is not defined in `passwd.tab(5)`.
- ▶ Method not found  
MPA method defined by `nodehm.tab(5)` not found in `$XCATPREFIX/lib`.

- ▶ MPA type not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not ping MPA \$MPA  
\$MPA is the MPA defined in nodelist.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ mpcliroot not defined in site.tab  
See site.tab(5).
- ▶ MPA command not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not HTTP to MPA \$MPA  
\$MPA is the MPA defined in nodelist.tab(5). Returned if HTTP reset fails. PCI MPA may be rebooting or locked up. **rreset(1)** if MPA is powered by an APC MasterSwitch.

## Examples

```
[root]# mpareset asma1  
asma1: reset
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **mpacheck(1)**, **mpascan(1)**, **mpasetup(1)**, **mpncheck(1)**, **rreset(1)**

# mpascan - Scan MPA for RS485 chained nodes

## Synopsis

```
mpascan noderange  
mpascan [-h|--help|-v|--version]
```

## Description

**mpascan** displays the list of internal management processor node names associated with a single or range of MPAs scanned. MPAs must be listed as nodes in `nodelist.tab(5)` and in `/etc/hosts` and/or DNS. MPAs must be defined in `mpa.tab(5)`.

## Options

<b>noderange</b>	See <code>noderange(3)</code> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b><code>\$XCATROOT/etc/nodelist.tab</code></b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/mpa.tab</code></b>	xCAT management processor file. See <code>mpa.tab(5)</code> for further details.
<b><code>/etc/hosts</code></b>	System host table.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
MPA(s) are not defined in `nodelist.tab(5)`.
- ▶ `asmauser` not defined in `passwd.tab`  
Default user name is not defined in `passwd.tab(5)`.
- ▶ `asmapass` not defined in `passwd.tab`  
Default MPA password is not defined in `passwd.tab(5)`.
- ▶ Method not found  
MPA method defined by `nodehm.tab(5)` not found in `$XCATPREFIX/lib`.



- ▶ MPA type not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not **ping** MPA \$MPA  
\$MPA is the MPA defined in nodelist.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ mpcliroot not defined in site.tab  
See site.tab(5).
- ▶ MPA command not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not **telnet** to MPA \$MPA, and other telnet errors  
\$MPA is the MPA defined in nodelist.tab(5). Returned if telnet port check fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA or **rreset(1)** if MPA is powered by an APC MasterSwitch.

## Examples

```
[root]# mpascan asma1
asma1: node1 5
asma1: node2 4
asma1: node3 1
asma1: node4 2
asma1: RJB12354343 3
```

Column one is the MPA, column two is the internal management processor name on the RS485 chain, and column three is the internal dynamically assigned index number for the RS485 chain.

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **mpacheck(1)**, **mpareset(1)**, **mpasetup(1)**, **mpncheck(1)**, **rreset(1)**

# mpasetup - Set MPA settings

## Synopsis

```
mpasetup noderange  
mpasetup [-h|--help|-v|--version]
```

## Description

**mpasetup** validates that xCAT can communicate with a single or range of IBM MPAs. MPAs must be listed as nodes in `nodelist.tab(5)`, `mpa.tab(5)`, and in `/etc/hosts` and/or DNS. **mpasetup** also returns the IP configuration of the MPA.

## Options

<b>noderange</b>	See <code>noderange(3)</code> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b><code>\$XCATROOT/etc/nodelist.tab</code></b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/mpa.tab</code></b>	xCAT management processor file. See <code>mpa.tab(5)</code> for further details.
<b><code>/etc/hosts</code></b>	System host table.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
MPA(s) are not defined in `nodelist.tab(5)`.
- ▶ `asmauser` not defined in `passwd.tab`  
Default user name is not defined in `passwd.tab(5)`.
- ▶ `asmapass` not defined in `passwd.tab`  
Default MPA password is not defined in `passwd.tab(5)`.
- ▶ Method not found  
MPA method defined by `nodehm.tab(5)` not found in `$XCATPREFIX/lib`.

- ▶ MPA type not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not ping MPA \$MPA  
\$MPA is the MPA defined in nodelist.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ mpcliroot not defined in site.tab  
See site.tab(5).
- ▶ MPA command not defined in mpa.tab  
See mpa.tab(5).
- ▶ Could not telnet to MPA \$MPA, and other telnet errors  
\$MPA is the MPA defined in nodelist.tab(5). Returned if telnet port check fails. PCI MPA may be in use or telnet interface is locked up. `mpareset(1)` can be used to reboot the PCI MPA or `rreset(1)` if MPA is powered by an APC MasterSwitch.

## Examples

```
[root]# mpasetup asma1
asma1: SUCCESS: setsnmp -agent type=boolean true
asma1: SUCCESS: setsnmp -traps type=boolean true
asma1: SUCCESS: setsnmp -communityname type=int type=String 1 public
asma1: SUCCESS: setsnmp -ipaddress type=int type=int type=String 1 1
199.88.179.22
asma1: SUCCESS: setmpid -text type=String asma1
asma1: SUCCESS: setmpid -numeric type=String 10001
asma1: SUCCESS: setnethw -duplex type=duplexStr:AUTO,FULL,HALF HALF
asma1: SUCCESS: setnethw -datarate type=dataRateStr:AUTO,4M,10M,16M,100M
100M
asma1: SUCCESS: setalertrigger -enabled type=String noncritical.voltage
asma1: SUCCESS: setalertrigger -enabled type=String
noncritical.temperature
asma1: SUCCESS: setalertrigger -enabled type=String
noncritical.single_fan
asma1: SUCCESS: setalertrigger -enabled type=String noncritical.rps
asma1: SUCCESS: setalertrigger -enabled type=String
noncritical.expansion_device
asma1: SUCCESS: setalertrigger -enabled type=String critical.vrm
asma1: SUCCESS: setalertrigger -enabled type=String critical.voltage
asma1: SUCCESS: setalertrigger -enabled type=String critical.temp
asma1: SUCCESS: setalertrigger -enabled type=String critical.tamper
asma1: SUCCESS: setalertrigger -enabled type=String critical.power_supply
asma1: SUCCESS: setalertrigger -enabled type=String critical.multiple_fan
asma1: SUCCESS: setalertrigger -enabled type=String critical.dasd
```

```
asma1: SUCCESS: setalertrigger -enabled type=String system.post
asma1: SUCCESS: setalertrigger -enabled type=String system.os
asma1: SUCCESS: setalertrigger -enabled type=String system.loader
asma1: SUCCESS: setalertrigger -enabled type=String system.application
asma1: SUCCESS: setalertrigger -enabled type=String system.power_off
asma1: SUCCESS: setalertrigger -enabled type=String system.power_on
asma1: SUCCESS: setalertrigger -enabled type=String system.boot
asma1: SUCCESS: setalertrigger -enabled type=String system.pfa
asma1: Restarting MP, please wait...
asma1: SUCCESS: restartmp -flag default
asma1: PASSED: The management processor has been successfully restarted
```

## Author

Egan Ford (egan@us.ibm.com).

## Bugs

Unknown.

## See also

**noderange(3), mpareset(1), mpasetup(1), mpascan(1), mpncheck(1), rreset(1)**

# node1s - List node properties from tables

## Synopsis

```
node1s [noderange] [group|pos|type|rg|install|hm|all]
node1s [noderange] hm.{power|reset|cad|vitals|inv|cons}
                    hm.{bioscons|eventlogs|getmacs|netboot}
                    hm.{eth0|gcons|all}
node1s [noderange] rg.{tftp|nfs_install|install_dir|serial}
                    rg.{usenis|install_roll|acct|gm|pbs}
                    rg.{access|gpfs|ksdevice|all}nodeset
node1s [-h|--help|-v|--version]
```

## Description

**node1s** lists the node characteristics provided by `nodelist.tab`, `nodetype.tab`, `nodehm.tab`, and `noderes.tab`. Listings are organized by which table is used as the source of information.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>group pos</b>	Get information for specific fields from <code>nodelist.tab</code> .
<b>type</b>	Get information for specific fields from <code>nodetype.tab</code> .
<b>install</b>	Combine files from <code>rg</code> and <code>type</code> to get the install Kickstart files that are generated by xCAT.
<b>all</b>	Get information for all fields from all tables.
<b>hm</b>	Equivalent to <code>hm.all</code> . Get information for specific fields from <code>nodehm.tab</code> .
<b>rg</b>	Get the node resource group from <code>noderes.tab</code> . Get information for specific fields from <code>noderes.tab</code> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Author

Egan Ford ([egan@us.ibm.com](mailto:egan@us.ibm.com)).

# noderange - Generate a list of node names

## Synopsis

**noderange** is defined as a set of [comma delimited nodelists and grouplists]

## Description

The **noderange** variable is the principle argument of the xCAT commands. **noderange** is a function in `$XCATROOT/lib/functions` that is called internally by most xCAT commands so that a single operation may be applied to a range of nodes, often in parallel.

**noderange** lists can be:

- ▶ An individual node and/or group:

```
node01
group1
```

- ▶ A range of nodes and/or groups:

```
node01-node10
group1-group3
```

- ▶ A regular expression match of nodes and/or groups:

```
@node[345].*
@group[12].*
```

- ▶ An incremented range of nodes:

```
node10+5
```

- ▶ A file containing noderanges of nodes and/or groups:

```
^/tmp/nodelist
```

- ▶ A node shorthand range of nodes:

```
10-20
10+5
```

- ▶ Or any combination:

```
node01-node30,node40,^/tmp/nodes,@node[13].*,2-10,node50+5
```

Any individual noderange may be prefixed with an exclusion operator (default -) with the exception of the file operator (default ^).

Any combination or multiple combinations of inclusive and exclusive ranges of nodes and groups is legal. There is no precedence implied in the order of the arguments. Exclusive ranges have precedence over inclusive.

Nodes have precedence over groups. If a node range match is made then no group range match will be attempted.

All node names are validated against `nodelist.tab(5)`. Invalid nodes are ignored and return nothing.

All group names are validated against `nodelist.tab(5)`, `nodetype.tab(5)`, and `nodemodel.tab(5)`. Invalid groups are ignored and return nothing.

Throughout this man page the term xCAT node format will be used.

xCAT node format is defined by the following regex:

```
^([A-Za-z][A-Za-z-]*)([0-9]+)([A-Za-z-][A-Za-z0-9-]*)
```

In plain English, a node or group is in xCAT node format if, starting from the beginning, there are one or more alpha characters of any case and any number of a hyphen (-) in any combination, followed by one or more numbers, then optionally followed by one alpha character of any case or the hyphen followed by any combination of case-mixed alpha numerics and the hyphen.

**noderange** supports node/group names in any format. xCAT node format is not required; however, some node range methods used to determine range will not be used. For example, if using a **noderange** of `node1a-node9a` with a `nodelist.tab(5)` only listing `node1a` through `node5a`, **noderange** will enumerate then validate and return a proper range. If using a node range of `aa-az` with `nodelist.tab(5)` only listing `aa` through `ay`, **noderange** will fail to return any values.

Example xCAT node format node/group names are:

```
noderange prefix number suffix
node1 node 1
node001 node 001
node-001 node- 001
node-foo-001-bar node-foo- 001 -bar
node-foo-1bar node-foo- 1 bar
foo1bar2 foo 1 bar2
rack01unit34 rack 01 unit34
unit34rack01 unit 34 rack01
pos0134 pos 0134
```

Example non-xCAT node format node/group names, but still valid names, are:

```
aa
yellow
red
```

**noderange** will use multiple methods to generate node names:

1. **noderange** checks for the multiple range operator (default ,). Each range is also processed by **noderange**.
2. **noderange** checks for the file operator (default ^). If the file exists each line will be processed as a **noderange**. Lines starting with # or the file operator (default ^) are ignored. Only one **noderange** per line is read. All characters are ignored after the first white space.

For example,

```
~/tmp/nodes
```

Where:

```
[root]# cat /tmp/nodes outputs:  
# my node list (this line is ignored)  
~/tmp/foo #ignored  
node01 #node comment  
node02  
node03  
node10-node20  
@group[456].*  
-node50
```

3. **noderange** checks for the exclusion operator (default -) then continues. This operator supports nodes and groups. **noderange** is smart and will not confuse the exclusion or range operators with the - character in names.
4. **noderange** checks for the for the regular expression operator (default @). Regular expressions offer the most flexibility.
5. **noderange** checks for a numeric-only range (for example, 10-20, 5+3, or just 10), then uses `$XCAT_NODE_PREFIX` and `$XCAT_NODE_SUFFIX` (optional) as the defaults to complete the node names. `$XCAT_NODE_PREFIX` must be defined to use **noderange** shorthand. If you use padded node names (for example, node001, node002, etc...) then you must specify `$XCAT_NODE_PADDING` or the default of 1 will be used. For example, if you use node names node001, node002, etc., then `$XCAT_NODE_PADDING` should be set to 3. **noderange** shorthand supports nodes only. **noderange** shorthand can be mixed with all other operators except regex (that is, exclusion, increment, range, and file may be used).
6. **noderange** checks for the increment range operator (default +). Increment range operator **noderanges** are in the format:

```
valid_node_name+number_of_sequential_nodes
```

For example, the following would yield node10 plus the next 4 nodes.

```
node10+5
```



This action is performed using two different methods. If `valid_node_name` is in xCAT node format then the range is enumerated to one less than `number_of_sequential_nodes`. If not in xCAT node format, then a sorted `nodelist.tab(5)` is used to return the node range. This operator supports nodes only.

7. **noderange** checks for a single node name or group name.
8. **noderange** checks for the range operator (default `-`). Ranges are performed first by validating that both the start and end nodes or groups defining the range exist and if so the range is returned based on the content of `nodelist.tab(5)`, `nodetype.tab(5)`, and `nodemodel.tab(5)`. If the start and end nodes or groups defined in the range do not exist, if both are in xCAT node format, and if both the prefix and suffix match, then the range is enumerated and each node/group validated. Only valid nodes/groups will be returned. **noderange** is smart and will not confuse the exclusion or range operators with the `-` character in names.
9. **noderange** returns nothing if no match can be found.

**noderange** uses the smallest integer to determine padding. For example, `node1–node10` will generate a list of nodes with numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. `node001–node010` will generate a list of nodes with numbers 001, 002, 003, 004, 005, 006, 007, 008, 009, 010.

Multiple instances of a node name are treated as one instance. For example, `node1–node10,node4,node4,node4` will generate a list of nodes numbered 1 through 10; the number 4 will only be listed once.

## Options

NA

## Environmental variables

**noderange** supports the following user-definable environmental variables to customize the behavior of **noderange**.

To change for all users, put variables in `/etc/profile.d/xcat.sh`.

<b>XCAT_NR_DELIMIT</b>	Defines the multiple range operator (default <code>,</code> )
<b>XCAT_NR_FILE</b>	Defines the range file operator (default <code>^</code> ).
<b>XCAT_NR_EXCLUDE</b>	Defines the exclude range operator (default <code>-</code> ).
<b>XCAT_NR_REGEX</b>	Defines the regular expression range operator (default <code>@</code> ).

<b>XCAT_NR_PLUS</b>	Defines the increment range operator (default +).
<b>XCAT_NR_RANGE</b>	Defines the range operator (default -).
<b>XCAT_NODE_PREFIX</b>	Defines the default node prefix used for shorthand node ranges (default node).
<b>XCAT_NODE_PADDING</b>	Defines the default node padding used for shorthand node ranges (default 1).
<b>XCAT_NODE_SUFFIX</b>	Defines the default node suffix used for shorthand node ranges (default <i>undefined</i> ).

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT node file used for node and group matches. See nodelist.tab(5) for further details.
<b>\$XCATROOT/etc/nodetype.tab</b>	xCAT node type file used for group matches. See nodetype.tab(5) for further details.
<b>\$XCATROOT/etc/nodemodel.tab</b>	xCAT node model file used for group matches. See nodemodel.tab(5) for further details.

## Example

```
all,-node5-node10
```

Generates a list of all nodes (assuming all is a group) listed in nodelist.tab(5) less node5 through node10.

```
node1-node10,-node3-node5,node4
```

Generates a list of nodes 1 through 10 less nodes 3, 4, 5.

Note that node4 is listed twice, first in the range and then at the end. Because exclusion has precedence node4 will be excluded.

```
node1-node10,-node3,-node5
```

Generates a list of nodes 1 through 10 less nodes 3 and 5.

```
-node17-node32,all
```

Generates a list of all (assuming all is a group) nodes in nodelist.tab(5) less 17 through 32.

```
node1-node128,user1-user4
```

Generates a list of nodes 1 through 128, and user nodes 1 through 4.

```
all,-rack1-rack3,-node100-node200,node150,-storage
```

Generates a list of all nodes (assuming all is a group), less nodes in groups rack1 through rack3 (assuming groups rack1, rack2, and rack3 are defined), less nodes 100 through 200, less nodes in the storage group. Note that node150 is listed explicitly, but is still excluded.

```
@node[23].*
```

Generates a list of nodes matching the regex `node[23].*`. That is, all nodes that start with `node2` or `node3` and end in anything or nothing. For example, `node2`, `node3`, `node20`, `node30`, `node21234` all match.

## Bugs/features

No match returns, no error.

## Author

Egan Ford (egan@us.ibm.com).

# nodeset - Set the boot state for a noderange

## Synopsis

```
nodeset [noderange] [boot|install|state|stat]
nodeset [noderange] [stage2|stage3|shell]
nodeset [noderange] [clone|cloneserver|image]
nodeset [noderange] [flash=image]
nodeset [-h|--help|-v|--version]
```

## Description

**nodeset** sets the next cold or warm boot state for a range of nodes or groups. **nodeset** accomplishes this by changing the network boot files. Each xCAT node always boots from the network and downloads a boot file with instructions on what action to take next.

**nodeset** only supports PXE Linux, Etherboot, and ELILO as network boot loaders. **nodeset** calls **nodeset.pxe**, **nodeset.eb**, and **nodeset.elilo** to perform the updates.

Assume that /tftpboot is the root for tftpd.

**nodeset.pxe** makes changes to /tftpboot/pxelinux.0/{node hex ip}.

**nodeset.eb** makes changes to /etc/dhcpd.conf.

**nodeset.elilo** makes changes to /tftpboot/elilo/{node ip}.conf.

**nodeset** only sets the state, but does not reboot.

**nodeset** is called by **rinstall** and **winstall** and is also called by the installation process remotely to set the boot state back to "boot".

When **nodeset** is called to set a node for an installation state, a NODETYPE-RESOURCE.BOOTTYPE template must exist in /tftpboot/xcat (or the appropriate directory as defined in site.tab(5)), where NODETYPE is defined per node in nodetype.tab(5), RESOURCE is defined per node or group in noderes.tab(5), and BOOTTYPE is defined per node in nodehm.tab(5).

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>boot</b>	Instruct node to boot local harddisk 0 on next boot.

<b>install</b>	Instruct node to boot from network. This usually involves TFTP downloading a kernel, initrd, and kernel options, then booting the downloaded images to facilitate automated unattended installation.
<b>state   stat</b>	Display the next boot state.
<b>stage2</b>	Instruct node to boot from network image stage2. This involves TFTP downloading a special prebuilt kernel, initrd, and kernel options used for MAC address collection. For a new node this is the default action and cannot be changed until a node entry exists in dhcpd.conf(5). It is not necessary to explicitly set stage2, unless it is used for testing and development purposes.
<b>stage3</b>	Instruct node to boot from network image stage3. This involves TFTP downloading a special prebuilt kernel, initrd, and kernel options used for the automated programming of IBM management processors.
<b>shell</b>	Instruct node to boot from a network maintenance shell image. This involves TFTP downloading a special prebuilt kernel, initrd, and kernel options used for a limited RAMdisk-based maintenance shell. Only available for Itanium-based systems.
<b>clone</b>	On reboot, instruct the node to get a cloned system image from the clone server.
<b>cloneserver</b>	On reboot, the node will become a clone server. The cloneserver supplies a system image clone for any node that requests it as a clone client.
<b>image</b>	On reboot, the node will create an image of itself on the master node. You can then use <b>rinstall</b> to install that image on all other nodes.
<b>flash=<i>image</i></b>	On reboot, the node will run remote flash using the specified image.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See nodelist.tab(5) for further details.
------------------------------------	--

<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the network boot type: PXE, Etherboot, ELILO, or NA.
<b>\$XCATROOT/etc/noderes.tab</b>	xCAT node resources file. See noderes.tab(5) for further details. This is used to determine the node's resource group.
<b>\$XCATROOT/etc/nodetype.tab</b>	xCAT node installation type file. See nodetype.tab(5) for further details. This is used to determine the node installation image type.
<b>\$XCATROOT/etc/site.tab</b>	xCAT main configuration file. See site.tab(5) for further details. This is used to determine the location of the TFTP root directory and the TFTP xCAT subdirectory. /tftpboot and /tftpboot/xcat is the default.
<b>/etc/dhcpd.conf</b>	xCAT dhcpd configuration file. See dhcpd.conf(5) for further details. This is used by <b>nodeset</b> to determine whether a node will only boot stage2 because no statically assigned IP exists for that node. Also used by <b>nodeset .eb</b> to set the boot state for Etherboot-enabled nodes.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group
  - A node or group is not defined in nodelist.tab(5).
- ▶ Must be run from \$MASTER
  - nodeset** must be run from the node defined in site.tab(5) as "master".
- ▶ Cannot write to \$TFTPDIR/\$TFTPXCATROOT dir
  - The tftpdir and tftpboot entries in site.tab(5) do not exist as directories.
- ▶ no \$TFTPDIR/\$TFTPXCATROOT dir
  - The tftpdir and tftpboot entries in site.tab(5) are not defined.
- ▶ **nodeset** function not supported
  - The netboot field in nodehm.tab(5) defined for that node is "NA".

- ▶ netboot method \$NETBOOT undefined  
The netboot field in nodehm.tab(5) is not defined.
- ▶ \$FTTPDIR/\$FTTPXCATROOT/\$NODETYPE-\$RESOURCE.\$NETBOOT does not exist  
Installation template file does not exist.
- ▶ No resources defined in \$NODERESTAB  
The node in question does not have a node or group entry in noderes.tab(5).
- ▶ No node type in \$NODETYPETAB  
The node in question does not have a entry in nodetype.tab(5).
- ▶ no dhcp entry  
The node in question does not have a entry in dhcpd.conf(5). This limits the node to stage2 until MACs has been collected and **makedhcp(8)** have been run.
- ▶ \$FLAG not implemented  
The node state has not been defined for that network boot type. For example, the shell state only available to ELILO (**nodeset.e11o**).
- ▶ \$FLAG function not supported  
You tried to pass an incorrect state to the back-end methods.

## Examples

```
[root]# nodeset node5 install
node5: install compute62-compute
```

Install node5 with the compute62 image using resources from the compute resource group.

## Bugs

Simultaneous **nodeset.eb** calls from remote hosts can corrupt /etc/dhcpd.conf(5).

Etherboot is not recommend for a large number of nodes.

## Author

Egan Ford (egan@us.ibm.com).

## See also

`noderange(3)`, `node1s(1)`, `nodeset(1)`, `rinstall(1)`, `winstall(1)`, `makedhcp(8)`



# pping - Parallel ping

## Synopsis

```
pping [-s] [noderange]
pping [-h|--help|-v|--version]
```

## Description

**pping** is a utility used to **ping** lists of nodes in parallel. See **noderange(3)**.

**pping** will return an unsorted list of nodes with a ping or noping status. The list is actually sorted by first **ping**, unless **-s** is specified.

**pping** front-ends **ping** and **fping** if available.

## Options

<b>-s</b>	<b>ping</b> serially.
<b>noderange</b>	See <b>noderange(3)</b> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

**\$XCATROOT/etc/nodelist.tab** xCAT nodelist file. See **nodelist.tab(5)** for further details.

## Diagnostics

The following diagnostic may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in **nodelist.tab(5)**.

## Examples

```
[root]# pping gpfs
node4: ping
node5: ping
node6: noping
```

## **Bugs**

Unknown.

## **Author**

Egan Ford (egan@us.ibm.com).

## **See also**

**noderange(3), prcp(1), prsync(1), psh(1)**

# prcp - Parallel remote copy

## Synopsis

```
prcp [filename filename ...] [noderange:destination directory]
prcp [-r] [filename filename ...] [directory directory...]
      [noderange:destination directory]
prcp [-h|--help|-v|--version]
```

## Description

**prcp** is a utility used to copy a single or multiple set of files and/or directories to a single or range of nodes and/or groups in parallel.

**prcp** is a front-end to the remote copy method defined in `site.tab(5)` (usually `scp` or `rcp`).

**prcp** does *not* multicast, but does use parallel unicasts.

## Options

<b>-r</b>	Recursive copy through directories.
<b>filename</b>	A space-delimited list of files to copy.
<b>directory</b>	A space-delimited list of directories to copy.
<b>noderange:destination</b>	A <b>noderange(3)</b> and destination directory. The <code>:</code> is required.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b>\$XCATROOT/etc/site.tab</b>	xCAT main configuration file. <b>prcp</b> uses this to determine the remote copy command to use. See <code>site.tab(5)</code> for further details.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in nodelist.tab(5).
- ▶ Specify one noderange  
Multiple noderanges specified. For example, node1-node10:/tmp  
node20-node22:/tmp.
- ▶ is not allowed  
Use node1-node10,node20-node22:/tmp as the noderange:destination  
directory.
- ▶ No source files specified  
No files specified to be copied.
- ▶ No destination directory specified  
No remote destination directory specified.
- ▶ noping  
Remote node is not up.
- ▶ ssh and rsh errors  
Verify that **ssh** and **rsh** are set up with no prompting and access to all nodes  
allowed.

## Examples

```
[root]# prcp -r /usr/local node1,node3:/usr/local  
prcp passwd group rack01:/etc
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **pping(1)**, **prsync(1)**, **psh(1)**

# prsync - parallel rsync

## Synopsis

```
prsync [filename filename ...] [noderange:destination directory]
prsync [rsync opts] [filename filename ...] [directory directory...]
      [noderange:destination directory]
prsync [-h|--help|-v|--version]
```

## Description

**prsync** is a frontend to **rsync** a range of nodes and/or groups in parallel.

**prsync** does *not* multicast, but does use parallel unicasts.

## Options

<b>rsync opts</b>	<b>rsync</b> options. See <b>rsync(1)</b> .
<b>filename</b>	A space-delimited list of files to <b>rsync</b> .
<b>directory</b>	A space-delimited list of directories to <b>rsync</b> .
<b>noderange:destination</b>	A <b>noderange(3)</b> and destination directory. The <b>:</b> is required.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/site.tab</b>	xCAT main configuration file. <b>prsync</b> uses this to determine the remote shell command to use. See <b>site.tab(5)</b> for further details.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in **nodelist.tab(5)**.

- ▶ Specify one noderange  
Multiple noderanges specified. For example, node1-node10:/tmp  
node20-node22:/tmp.
- ▶ Is not allowed.  
Use node1-node10,node20-node22:/tmp as the noderange:destination  
directory.
- ▶ No source files specified  
No files specified to be copied.
- ▶ No destination directory specified  
No remote destination directory specified.
- ▶ noping  
Remote node is not up.
- ▶ ssh and rsh errors  
Verify that **ssh** and **rsh** are set up with no prompting and access to all nodes  
is allowed. **rsync** requires **rsh** or **ssh**.
- ▶ rsync errors  
See **rsync(5)**.

## Examples

```
[root]# cd /install; prsync -crazv post stage:/install
prsync passwd group rack01:/etc
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **prcp(1)**, **pping(1)**, **psh(1)**

# psh - Parallel remote shell

## Synopsis

```
psh [-s] [noderange|me|pbs job id] [command]
psh [-h|--help|-v|--version]
```

## Description

**psh** is a utility used to run a command across a list of nodes in parallel. See **noderange(3)**.

**psh** relies on the **rsh** field in **site.tab(5)** to determine whether to use **rsh**, **ssh**, or any other method to launch a remote shell. **rsh**, **ssh**, or any other method must be set up to allow no prompting (that is, **.rhosts** for **rsh** and **sshd\_config** and **.rhosts** for **ssh**) for **psh** to work.

## Options

<b>-s</b>	Issues the commands serially.
<b>noderange</b>	See <b>noderange(3)</b> .
<b>me</b>	Run against nodes owned by "me" as listed by PBS's <b>qstat(1B)</b> command.
<b>pbs</b>	Run against nodes assigned to a PBS job as listed by PBS's <b>qstat(1B)</b> command.
<b>command</b>	Command to be run in parallel. If no command is given, then <b>psh</b> enters interactive mode. In interactive mode a > prompt is displayed. Any command entered is executed in parallel to the nodes in the noderange. Use Exit or Ctrl+D to end the interactive session.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/site.tab</b>	xCAT main configuration file. <b>psh</b> uses this to determine the remote shell command to use. See <b>site.tab(5)</b> for further details.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in nodelist.tab(5).
- ▶ noping  
Remote node is not up.
- ▶ ssh and rsh errors  
Verify that **ssh** and **rsh** are set up with no prompting and access to all nodes allowed.

## Examples

```
[root]# psh node4-node6 date
node4: Sun Aug 5 17:42:06 MDT 2001
node5: Sun Aug 5 17:42:06 MDT 2001
node6: Sun Aug 5 17:42:06 MDT 2001
[root]# psh node1-node3
Executing on: node1 node2 node3
[root]# psh all /sbin/halt
Shutdown down all nodes listed in nodelist.tab.
[root]# psh all 'grep processor /proc/cpuinfo | wc -l' | sort
Will return a list of nodes with the number of processors per node sorted
by node.
```

## Bugs

**ssh** issues.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **prcp(1)**, **pping(1)**, **prsync(1)**



# rcons - remote console

## Synopsis

```
rcons [singlenode]
rcons [-h|--help|-v|--version]
```

## Description

**rcons** provides access to a single remote node, *singlenode*, in a serial console.

## Options

<b>singlenode</b>	A valid node name from nodelist.tab(5).
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See nodelist.tab(5) for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the console access method.
<b>\$XCATROOT/etc/conserver.tab</b>	xCAT conserver file. See conserver.tab(5) for further details.
<b>\$XCATROOT/etc/rtel.tab</b>	xCAT reverse telnet file. See rtel.tab(5) for further details.
<b>\$XCATROOT/etc/tty.tab</b>	xCAT direct attached tty file. See tty.tab(5) for further details.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in nodelist.tab(5).
- ▶ Console function not supported  
Node does not support remote serial console.

- ▶ telnet errors  
`telnet` is called if the method is defined as `rte1` in `nodehm.tab(5)`.
- ▶ cu errors  
`cu` is called if the method is defined as `tty` in `nodehm.tab(5)`.
- ▶ conserver errors  
Console is called if the method is defined as `conserver` in `nodehm.tab(5)`.

## Examples

```
[root]# rcons node5
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

`rvid(1)`, `wcons(1)`, `wvid(1)`

# reventlog - Retrieve or clear remote hardware event logs

## Synopsis

```
reventlog [noderange] [number of entries|all|clear]
reventlog [-h|--help|-v|--version]
```

## Description

**reventlog** can display any number of remote hardware event log entries or clear them for a range of nodes. Hardware event logs are stored on each server's management processor.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>number of entries</b>	Retrieve n number of entries.
<b>all</b>	Retrieve all entries.
<b>clear</b>	Clear event logs.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/mp.tab</b>	Management processor network definition table. See <b>mp.tab(5)</b> for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See <b>nodehm.tab(5)</b> for further details. This is used to determine the remote event log method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
The node or group is not defined in **nodelist.tab(5)**.

- ▶ Function not supported  
The remote event log retrieval method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in mp.tab(5). The PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ telnet timeout  
The PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. The PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ name lookup failure  
Returned if not in DNS.
- ▶ node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# reventlog node4,node5 5
node4: SERVPROC I 09/06/00 15:23:33 Remote Login Successful User ID =
USERID[00]
node4: SERVPROC I 09/06/00 15:23:32 System spn1 started a RS485 connection
with us[00]
node4: SERVPROC I 09/06/00 15:22:35 RS485 connection to system spn1 has
ended[00]
node4: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User ID =
USERID[00]
node4: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection
with us[00]
node5: SERVPROC I 09/06/00 15:22:32 Remote Login Successful User ID =
USERID[00]
node5: SERVPROC I 09/06/00 15:22:31 System spn1 started a RS485 connection
with us[00]
```

```
node5: SERVPROC I 09/06/00 15:21:34 RS485 connection to system spn1 has
ended[00]
node5: SERVPROC I 09/06/00 15:21:30 Remote Login Successful User ID =
USERID[00]
node5: SERVPROC I 09/06/00 15:21:29 System spn1 started a RS485 connection
with us[00]
[root]# reventlog node4,node5 clear
node4: clear
node5: clear
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **mpacheck(1)**, **mpareset(1)**

# rinstall - Remote network install

## Synopsis

```
rinstall [noderange]
rinstall [-h|--help|-v|--version]
```

## Description

**rinstall** will force an unattended network install for a range of nodes. See **noderange(3)**. **nodeset(1)** is called to set the node install state, then **rpower(1)** noderange boot is called to force the server to boot. If a node is off it will be powered on. If a node is on it will be reset.

**rinstall** returns *nodename: install type*, where *install type* is defined by *nodetype.tab(5)* and *noderes.tab(5)*. It also returns *nodename: powerstate action*, where *powerstate* is on or off and *action* is on or reset for each node.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

**rinstall** is a simple frontend to **nodeset(1)** and **rpower(1)**. Check the **nodeset(1)** and **rpower(1)** man pages for files that **rinstall** requires.

## Diagnostics

**rinstall** is a simple frontend to **nodeset(1)** and **rpower(1)**. Check the **nodeset(1)** and **rpower(1)** man pages for diagnostics that may be issued on stdout/stderr.

## Examples

```
[root]# rinstall node4,node5
node4: compute73-compute
node5: stage73-compute
node4: off on
node5: on reset
```

node4 is defined as type compute73 with resources from the compute group, node5 is defined as type stage73 with resources from the compute group, node4 was off then powered on, and node5 was on then reset.

## **Bugs**

Unknown.

## **Author**

Egan Ford (egan@us.ibm.com).

## **See also**

**noderange(3), nodeset(1), rpower(1), winstall(1)**

# rinv - Remote hardware inventory

## Synopsis

```
rinv [noderange] [pci|config|model|serial|asset|vpd|bios|mprom|all]
rinv [-h|--help|-v|--version]
```

## Description

**rinv** retrieves hardware configuration information from the onboard management processor for a single or range of nodes and groups.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>pci</b>	Retrieves PCI bus information.
<b>config</b>	Retrieves number of processors, speed, total memory, and DIMM locations.
<b>model</b>	Retrieves model number.
<b>serial</b>	Retrieves serial number.
<b>asset</b>	Retrieves asset tag. Usually it is the MAC address of eth0.
<b>vpd</b>	Retrieves BIOS level and management processor firmware level.
<b>bios</b>	Retrieves BIOS level.
<b>mprom</b>	Retrieves management processor firmware level.
<b>all</b>	All of the above.
<b>-h   --help</b>	Prints help.
<b>-v   --version</b>	Prints version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/mp.tab</b>	management processor network definition table. See <b>mp.tab(5)</b> for further details.



**\$XCATROOT/etc/nodehm.tab** xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the remote inventory method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in nodelist.tab(5).
- ▶ Function not supported  
The remote power control method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in mp.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ telnet timeout  
PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with the MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# rinv node5 all
node5: Machine Type/Model 865431Z
node5: Serial Number 23C5030
```

```
node5: Asset Tag 00:06:29:1F:01:1A
node5: PCI Information
node5: Bus VendID DevID RevID Description Slot Pass/Fail
node5: 0 1166 0009 06 Host Bridge 0 PASS
node5: 0 1166 0009 06 Host Bridge 0 PASS
node5: 0 5333 8A22 04 VGA Compatible Controller 0 PASS
node5: 0 8086 1229 08 Ethernet Controller 0 PASS
node5: 0 8086 1229 08 Ethernet Controller 0 PASS
node5: 0 1166 0200 50 ISA Bridge 0 PASS
node5: 0 1166 0211 00 IDE Controller 0 PASS
node5: 0 1166 0220 04 Universal Serial Bus 0 PASS
node5: 1 9005 008F 02 SCSI Bus Controller 0 PASS
node5: 1 14C1 8043 03 Unknown Device Type 2 PASS
node5: Machine Configuration Info
node5: Number of Processors: 2
node5: Processor Speed: 866 MHz
node5: Total Memory: 512 MB
node5: Memory DIMM locations: Slot(s) 3 4
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

[noderange\(3\)](#), [mpacheck\(1\)](#), [mpareset\(1\)](#)

# rpower - Remote power control

## Synopsis

```
rpower [noderange] [on|off|stat|state|reset|boot|cycle]
rpower [-h|--help|-v|--version]
```

## Description

**rpower** controls the power for a range of nodes.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>on</b>	Turn power on.
<b>off</b>	Turn power off.
<b>stat   state</b>	Return the current power state.
<b>reset</b>	Send a hardware reset.
<b>boot</b>	If off, then power on. If on, then hard reset. This option is recommended over cycle.
<b>cycle</b>	Power off, then on.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/mp.tab</b>	Management processor network definition table. See <b>mp.tab(5)</b> for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See <b>nodehm.tab(5)</b> for further details. This is used to determine the remote power method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
Node or group is not defined in **nodelist.tab(5)**.

- ▶ Function not supported  
The remote power control method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in mp.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ telnet timeout  
PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with the MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# rpower node4,node5 stat
node4: on
node5: off
[root]# rpower node5 on
node5: on
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

`noderange(3)`, `mpacheck(1)`, `mpareset(1)`

# rreset - Remote hard reset

## Synopsis

```
rreset [noderange]
rreset [-h|--help|-v|--version]
```

## Description

**rreset** resets a range of nodes.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <b>nodelist.tab(5)</b> for further details.
<b>\$XCATROOT/etc/mp.tab</b>	Management processor network definition table. See <b>mp.tab(5)</b> for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See <b>nodehm.tab(5)</b> for further details. This is used to determine the remote hard reset method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
A node or group is not defined in **nodelist.tab(5)**.
- ▶ Function not supported  
The remote hard reset control method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in **mp.tab(5)**. PCI MPA adapter may not have power. Check Ethernet cable and IP.

- ▶ telnet timeout  
PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# rreset node3-node5
node3: reset
node4: reset
node5: reset
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **mpacheck(1)**, **mpareset(1)**

# rvid - Remote video (VGA)

## Synopsis

```
rvid [singlenode] [boot|nboot]
rvid [-h|--help|-v|--version]
```

## Description

**rvid** redirects the VGA text video from *singlenode* to your local tty. Useful for debugging boot issues. **rvid** is not intended for use as a general purpose remote OS console. Use **rcons(1)** and **wcons(1)** instead.

**Note:** Three things to be aware of are:

- ▶ While viewing remote video, no other management processor functions may be performed to any other node sharing the same management processor network.
- ▶ You must use Ctrl+X to exit so that the management processor can clean up. Failure to do so may lead to a hung SP (rare). To correct a hung SP, you will have to remove the power cord from the server and count to 15 (slowly).
- ▶ **rvid** is not supported for RSA devices. In general, remote console redirection, done through the BIOS settings, is preferred.

## Options

<b>singlenode</b>	A valid node name from <code>nodelist.tab(5)</code> .
<b>boot</b>	Forces the reset or power on before redirecting video.
<b>noboot</b>	Does not force the reset or power on before redirecting video. Default action if no option is given.
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b>\$XCATROOT/etc/mp.tab</b>	management processor network definition table. See <code>mp.tab(5)</code> for further details.



**\$XCATROOT/etc/nodehm.tab** xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the remote video method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
A node or group is not defined in nodelist.tab(5).
- ▶ Function not supported  
A node does not support remote hard reset control or the remote hard reset control method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in mp.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ telnet timeout  
PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# rvid node4 noboot
```

## **Bugs**

Unknown.

## **Author**

Egan Ford (egan@us.ibm.com).

## **See also**

**rcons(1), wvid(1), wcons(1)**

# rvitals - Remote hardware vitals

## Synopsis

```
rvitals [noderange] [cputemp|disktemp|ambtemp|temp]
rvitals [noderange] [voltage|fanspeed|power|powertime]
rvitals [noderange] [reboots|state|all]
rvitals [-h|--help|-v|--version]
```

## Description

**rvitals** retrieves vital hardware information from the onboard management processor for a range of nodes or groups.

## Options

<b>noderange</b>	See <b>noderange(3)</b> .
<b>cputemp</b>	Retrieves CPU temperatures.
<b>disktemp</b>	Retrieves HD back plane temperatures.
<b>ambtemp</b>	Retrieves ambient temperatures.
<b>temp</b>	Retrieves all temperatures.
<b>voltage</b>	Retrieves power supply and VRM voltage readings.
<b>fanspeed</b>	Retrieves fan speeds.
<b>power</b>	Retrieves power status.
<b>powertime</b>	Retrieves total power uptime. This value only increases, unless the management processor flash gets updated.
<b>reboot</b>	Retrieves total number of reboots. This value only increases, unless the management processor flash gets updated.
<b>state</b>	Retrieves the system state.
<b>all</b>	All of the above.
<b>-h   --help</b>	Prints help.
<b>-v   --version</b>	Prints version.

## Files

**\$XCATROOT/etc/nodelist.tab** xCAT nodelist file. See **nodelist.tab(5)** for further details.

<b>\$XCATROOT/etc/mp.tab</b>	Management processor network definition table. See mp.tab(5) for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the remote vitals method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
A node or group is not defined in nodelist.tab(5).
- ▶ Function not supported  
A node does not support remote hard reset control or the remote hard reset control method is not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network defined in mp.tab(5). PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ telnet timeout  
PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check cabling and try **mpareset(1)**.
- ▶ Remote session timeout  
Communication with the MPN was successful but communication with node failed. Check cabling. Try removing power from node, wait 10 seconds, then restore.

## Examples

```
[root]# rvitals node5 all
node5: CPU 1 Temperature: + 29.00 C (+ 84.2 F)
node5: CPU 2 Temperature: + 19.00 C (+ 66.2 F)
node5: DASD Sensor 1 Temperature: + 32.00 C (+ 89.6 F)
node5: System Ambient Temperature Temperature: + 26.00 C (+ 78.8 F)
node5: +5V Voltage: + 5.01V
node5: +3V Voltage: + 3.29V
node5: +12V Voltage: + 11.98V
node5: +2.5V Voltage: + 2.52V
node5: VRM1 Voltage: + 1.61V
node5: VRM2 Voltage: + 1.61V
node5: Fan 1 Percent of max: 100%
node5: Fan 2 Percent of max: 100%
node5: Fan 3 Percent of max: 100%
node5: Fan 4 Percent of max: 100%
node5: Fan 5 Percent of max: 100%
node5: Fan 6 Percent of max: 100%
node5: Current Power Status On
node5: Power On Seconds 11855915
node5: Number of Reboots 930
node5: System State Booting OS or in unsupported OS
```

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **mpacheck(1)**, **mpareset(1)**

# wcons - Windowed remote console

## Synopsis

```
wcons    [{-t|-tile|--tile} n | {-t|-tile|--tile}=n]  
         [{-f|-font|--font} font | {-f|-font|--font}=font]  
         noderange  
wcons    [-h|--help|-v|--version]
```

## Description

**rvitals** provides access to the remote node serial consoles of a range of nodes or groups. As the number of nodes requested increases the windows will get smaller.

**rvitals** is a simple frontend to **rcons** in an **xterm** session for each console.

**Note:** Use **wkll** to get rid of all open **wcons** windows.

Also, to change the font size in a window: Shift+right-click.

## Options

<b>-tl-tile --tile <i>n</i></b>	Tile <b>wcons(1)</b> windows from top left to bottom right. If <i>n</i> is specified then tile <i>n</i> across. If <i>n</i> is not specified then tile to edge of screen. If tiled <b>wcons(1)</b> windows reach bottom right, then the windows start at top left overlaying existing <b>wcons(1)</b> windows.
<b>-fl-font --font <i>font</i></b>	Use X Windows font <i>font</i> for <b>wcons</b> windows. <b>wcons</b> supports the following font aliases: verysmall   vs = nil2 smallls = 5x8 medium   medlm = 6x13 large   l   big   b = 7x13 verylarge   vl   verybig   vb = 10x20
<b>noderange</b>	See <b>noderange(3)</b> .
<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

<b><code>\$XCATROOT/etc/nodelist.tab</code></b>	xCAT nodelist file. See <code>nodelist.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/nodehm.tab</code></b>	xCAT node hardware management file. See <code>nodehm.tab(5)</code> for further details. This is used to determine the console access method.
<b><code>\$XCATROOT/etc/conserver.tab</code></b>	xCAT conserver file. See <code>conserver.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/rtel.tab</code></b>	xCAT reverse telnet file. See <code>rtel.tab(5)</code> for further details.
<b><code>\$XCATROOT/etc/tty.tab</code></b>	xCAT direct attached tty file. See <code>tty.tab(5)</code> for further details.

## Diagnostics

The following diagnostics may be issued on `stdout/stderr`:

- ▶ Is not a node or a group  
A node or group is not defined in `nodelist.tab(5)`.
- ▶ Console function not supported  
Node does not support remote serial console.
- ▶ telnet errors  
**telnet** is called if the method is defined as `rtel` in `nodehm.tab(5)`.
- ▶ cu errors  
**cu** is called if the method is defined as `tty` in `nodehm.tab(5)`.
- ▶ Conserver errors  
`console` is called if the method is defined as `conserver` in `nodehm.tab(5)`.
- ▶ No DISPLAY  
DISPLAY environment variable is not set.
- ▶ xterm errors  
**wcons** calls **xterm**, so standard X11/xterm errors may occur.

## Examples

See Figure A-1.

```
[root]# wcons node1-node5
```

```
[root]# wcons --tile --font=nil2 all
[root]# wcons -t 4 node1-node16
[root]# wcons -f vs -t 4 node1-node4
[root]# wcons node001-node010
```

(Nodes 4 and 5 are installing. The rest of the nodes are at a login)

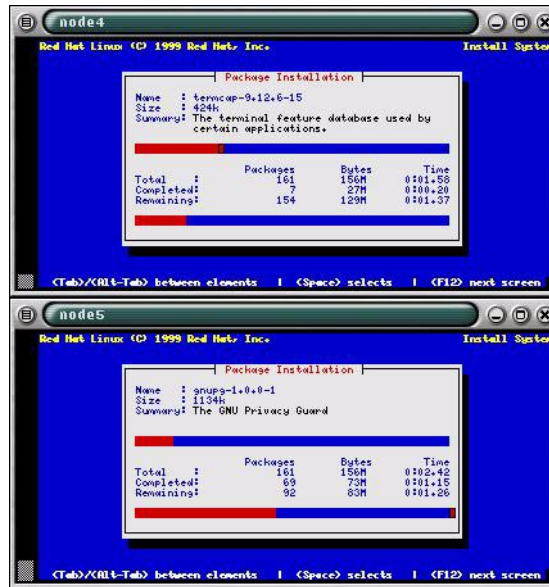


Figure A-1 Windowed remote console

## Bugs

Tile mode assumes that the width of the left window border is also the width of the right and bottom window border. Most window managers should not have a problem.

## Author

Egan Ford (egan@us.ibm.com).

## See also

**noderange(3)**, **rcons(1)**, **rvid(1)**, **wvid(1)**, **wkill(1)**



# winstall - Windowed remote network install

## Synopsis

```
winstall [{-t|-tile|--tile} n | {-t|-tile|--tile}=n]
        [{-f|-font|--font} font | {-f|-font|--font}=font]
        noderange
winstall [-h|--help|-v|--version]
```

## Description

**winstall** will force an unattended network install for a range of nodes or groups. See **noderange**(3).

When **winstall** is run, **wcons**(1) is called to open a remote serial console session in an **xterm**. Next, **nodeset**(1) is called to set the node install state, then **rpower**(1) **noderange** boot is called to force the server to boot. If a node is off it will be powered on. If a node is on it will be reset.

**winstall** returns *nodename: install type*, where *install type* is defined by **nodetype.tab**(5) and **noderes.tab**(5). It also returns *nodename: powerstate action*, where *powerstate* is on or off and *action* is on or reset for each node.

**winstall** is functionally equivalent to **rinstall**, except that **wcons** will launch an **xterm** window for the remote console display. **winstall** passes its arguments to **rinstall** and **wcons**.

## Options

- |                             |  |
|-----------------------------|--|
| <b>-tl-tile --tile n</b>    | Tile <b>wcons</b> (1) windows from top left to bottom right. If <i>n</i> is specified then tile <i>n</i> across. If <i>n</i> is not specified then tile to edge of screen. If tiled <b>wcons</b> (1) windows reach bottom right, then the windows start at top left overlaying existing <b>wcons</b> (1) windows.  |
| <b>-fl-font --font font</b> | Use X Windows font <i>font</i> for <b>wcons</b> windows. <b>wcons</b> supports the following font aliases:<br><small>verysmall</small>   <small>vs</small> = <small>nil2</small><br><small>smallls</small> = <small>5x8</small><br><small>medium</small>   <small>medlm</small> = <small>6x13</small><br><small>large</small>   <small>l</small>   <small>big</small>   <small>b</small> = <small>7x13</small><br><small>verylarge</small>   <small>vl</small>   <small>verybig</small>   <small>vb</small> = <small>10x20</small> |
| <b>noderange</b>            | See <b>noderange</b> (3).  |

<b>-h   --help</b>	Print help.
<b>-v   --version</b>	Print version.

## Files

**winstall** is a simple frontend to **wcons(1)**, **nodeset(1)**, and **rpower(1)**. Check the **wcons(1)**, **nodeset(1)**, and **rpower(1)** man pages for files that **winstall** requires.

## Diagnostics

**winstall** is a simple frontend to **wcons(1)**, **nodeset(1)**, and **rpower(1)**. Check the **wcons(1)**, **nodeset(1)**, and **rpower(1)** man pages for diagnostics that may be issued on stdout/stderr.

## Examples

```
[root]# winstall -t node4,node5
node4: compute73-compute
node5: stage73-compute
node4: off on
node5: on reset
```

node4 is defined as type compute73 with resources from the compute group, node5 is defined as type stage73 with resources from the compute group, node4 was off then powered on, node5 was on then reset. See Figure A-2.

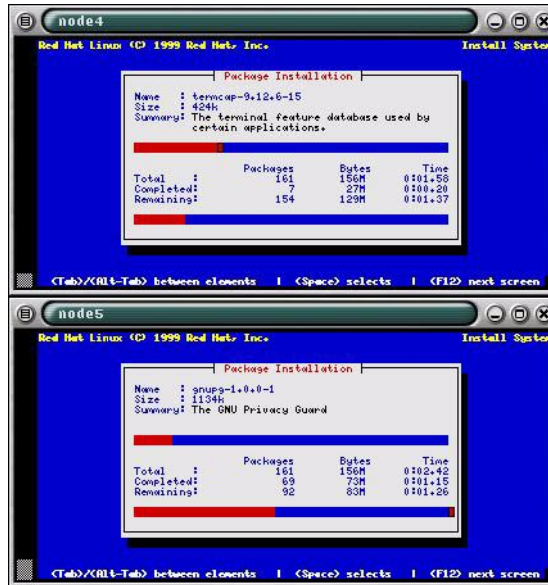


Figure A-2 Windowed remote network install

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

[noderange\(3\)](#), [nodeset\(1\)](#), [rinstall\(1\)](#), [rpower\(1\)](#)

# wkill - Windowed remote console kill

## Synopsis

```
wkill [noderange]
wkill [-h|--help|-v|--version]
```

## Description

**wkill** will kill the **wcons** windows on your **\$DISPLAY** for a single or range or nodes or groups.

**wkill** will only kill windows on your display and for only the **noderange(3)** you specify. If no **noderange(3)** is specified, then all **wcons** windows on your **\$DISPLAY** will be killed.

## Options

```
-h | --help          Print help.
-v | --version       Print version.
```

## Files

**\$XCATROOT/etc/nodelist.tab** xCAT nodelist file. See [nodelist.tab\(5\)](#) for further details.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
A node or group is not defined in [nodelist.tab\(5\)](#).
- ▶ No DISPLAY  
DISPLAY is not set.
- ▶ No **wcons** windows on **\$DISPLAY**  
Could not find **wcons** windows on **\$DISPLAY** to kill. Often happens when you **ssh** around--you forget where you were when you ran **wcons**.

## Examples

```
[root]# wkill node1-node5
[root]# wkill
```

The first example kills the first 5 nodes' consoles. The second example kills all open consoles.

## **Bugs**

Unknown.

## **Author**

Egan Ford (egan@us.ibm.com).

## **See also**

**noderange(3)**, **wcons(1)**.

# wvid - Windowed remote video (VGA)

## Synopsis

```
wvid [singlenode] [boot|nboot]
wvid [-h|--help|-v|--version]
```

## Description

**wvid** is a simple front-end to **rvid(5)**, except the **wvid** will launch an **xterm** with a geometry of 80x25 and use the VGA font if available.

**wvid** is not intended as a general purpose remote OS console; use **rcons(1)**, and **wcons(1)**.

**wvid** is slower than **wcons**, is not always interactive, and operates on a single node.

**Note:** There are a few things to take note of:

- ▶ While viewing remote video, no other management processor functions may be performed to any other node on the same management processor network.
- ▶ You must use Ctrl+X to exit so that the management processor can clean up. Failure to do so may lead to a hung SP (rare). To correct a hung SP, you will have to remove the power cord from the server and count to 15 (slowly).
- ▶ **wvid** is not supported for RSA devices, because **rvid** is not supported. In general, remote console redirection, done through the BIOS settings, is preferred.

## Options

<b>singlenode</b>	A valid node name from <code>nodelist.tab(5)</code> .
<b>boot</b>	Forces the reset or power on before redirecting video.
<b>noboot</b>	Does not force the reset or power on before redirecting video. Default action if no option is given.
<b>-h   --help</b>	Prints help.
<b>-v   --version</b>	Prints version.

## Files

<b>\$XCATROOT/etc/nodelist.tab</b>	xCAT nodelist file. See nodelist.tab(5) for further details.
<b>\$XCATROOT/etc/mp.tab</b>	Management processor network definition table. See mp.tab(5) for further details.
<b>\$XCATROOT/etc/nodehm.tab</b>	xCAT node hardware management file. See nodehm.tab(5) for further details. This is used to determine the remote video method.

## Diagnostics

The following diagnostics may be issued on stdout/stderr:

- ▶ Is not a node or a group  
A node or group is not defined in nodelist.tab(5).
- ▶ Function not supported  
A node does not support remote power control or remote power control method not supported.
- ▶ Ping to \$MPN failed  
\$MPN is the management processor network as defined in mp.tab(5). The PCI MPA adapter may not have power. Check the Ethernet cable and IP.
- ▶ telnet timeout  
The PCI MPA adapter may not have power. Check Ethernet cable and IP.
- ▶ Connection refused  
Returned if **telnet** fails. PCI MPA may be in use or telnet interface is locked up. **mpareset(1)** can be used to reboot the PCI MPA.
- ▶ Unknown asma \$host  
Returned if a node is not in /etc/hosts or DNS.
- ▶ Name lookup failure  
Returned if a node is not in DNS.
- ▶ Node not on asma \$MPN  
\$MPN is the MPN defined for that node in mp.tab(5). Use **mpacheck(1)** \$MPN to verify. Check the cabling and try **mpareset(1)**.

- ▶ Remote session timeout

Communication with the MPN is successful but communication with the node failed. Check cabling. Try removing power from the node, wait 10 seconds, then restore.

## Example

See Figure A-3.

```
[root]# wwid node5 boot
```

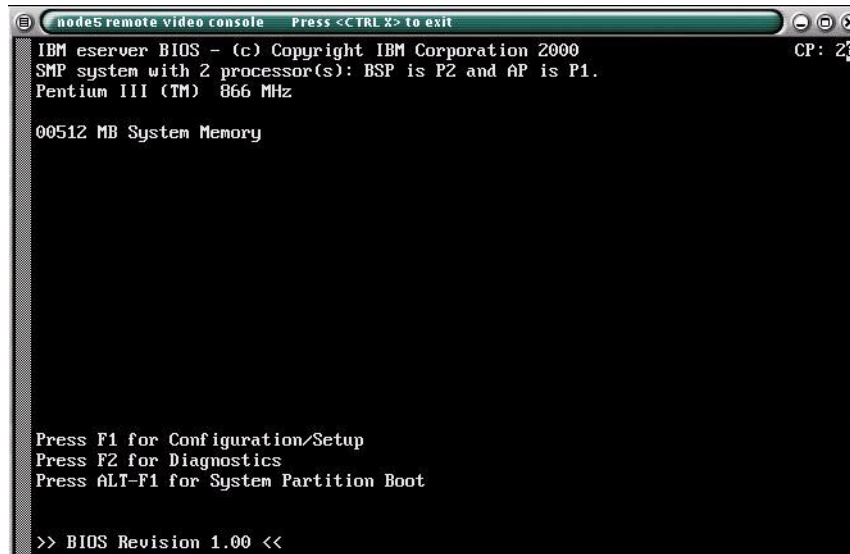


Figure A-3 Windowed remote video (VGA)

## Bugs

Unknown.

## Author

Egan Ford (egan@us.ibm.com).

## See also

rcons(1), rvidrvid(1), wcons(1)



**B**

## xCAT configuration tables

This appendix describes the tables used by xCAT during the configuration and installation process. These tables must be updated before starting the nodes installation.

Table B-1 provides a quick reference of the xCAT tables. Next in this appendix, we provide a full description of each xCAT table.

*Table B-1 xCAT tables description*

<b>Table</b>	<b>Description</b>
site.tab	This table is used to configure the cluster network. It includes a list of parameters that have to be updated for your cluster configuration.
nodelist.tab	This table describes all the nodes and the groups that they belong to.
nodes.tab	This table describes how each group of compute nodes can find the services it needs to boot and install resources.
nodetype.tab	This table contains one line per node that describes the name of the Kickstart file to use for the installation of that node.

<b>Table</b>	<b>Description</b>
nodehm.tab	This table defines the hardware management method used for each node and other managed devices. Because the hardware varies between clusters, this file gives flexibility to xCAT commands to adapt themselves to your environment.
mp.tab	This table describes the MPN topology. This allows xCAT to contact management processors through the associated MPA daisy chain.
mpa.tab	This table lists the MPAs in this cluster.
cisco3500.tab	This table describes the network links between the cluster nodes and the Cisco switch ports.
mac.tab	This table will be generated by the getmacs script. It contains one line per node. For each node we have its MAC address, which is used for the management for that node.
passwd.tab	This table will be used during the installation process and to perform some operations on the nodes, like power off or power on via the ASMA card.
apc.tab or apcp.tab	This table gives xCAT commands the ability to power on or power off pieces of equipment that are not controlled by the MPA card (network switch, Equinox, Myrinet switch, etc.). It works in conjunction with the APC MasterSwitch (apc.tab) or MasterSwitch Plus (apcp.tab) to control the electrical power on specified hardware.
conserver.tab	This table is used to map console servers to nodes. It is used only if conserver is being used to manage these serial ports.

Table	Description
rtel.tab	This table is used to map serial ports to nodes when using a terminal server that uses a reverse telnet model (Equinox ELS or iTouch). It is used only if conserver is <i>not</i> being used to manage these serial ports.
tty.tab	This table is used to map serial ports to nodes when using a terminal server that uses a tty model (Equinox ESP). It is used only if conserver is <i>not</i> being used to manage these serial ports.

## site.tab

This table is used to configure the cluster network. It includes a list of parameters that have to be updated for your cluster configuration.

Table B-2 Definition of site.tab parameters

Parameter	Description	Possible values
rsh	Command used to open a connection on a compute node.	/usr/bin/ssh
rscp	Command used to do a remote copy.	/usr/bin/scp
gkhfile	The file that will contain the ssh host keys for the nodes.	/opt/xcat/etc/gkh
sshkeyver	SSH protocol version. In order to use SSH protocol Version 2 you should use the SSH Release 3.	1 or 2
tftpdirc	Directory used by the TFTP daemon.	/tftpboot
tftpxcatroot	Directory under tftpdirc used for xCAT files.	xcat
domain	DNS domain used for the cluster.	clusters.com
dnssearch	Domain list to search when resolving DNS queries.	clusters.com
nameservers	List of DNS name server addresses.	10.0.0.1,192.168.0.254
forwarders	IP address of DNS server to forward request to, instead of doing recursive lookups.	10.42.69.211

Parameter	Description	Possible values
nets	This is where the network:netmask is used by the nodes for the cluster management virtual local area network (VLAN). Only this VLAN will be resolved by the DNS sever installed on the management node.	10.0.0.0:255.255.0.0, 10.1.0.0:255.255.0.0, 10.2.0.0:255.255.0.0, 192.168.42.0:255.255.255.0
dnsdir	Directory for DNS files.	/var/named
dnsallowq	List of DNS network:netmask pairs. This list determines the access permissions for DNS servers contained in the cluster.	10.0.0.0:255.255.0.0, 10.1.0.0:255.255.0.0, 10.2.0.0:255.255.0.0, 192.168.42.0:255.255.255.0
domainaliasip	IP address aliased to the cluster DNS domain name.	NA
mxhosts	This is used by DNS to identify FQDN mail servers.	NA
mailhosts	List of mail hosts for this cluster that will have a mailhost alias.	NA
master	Name of the master node.	masternode
homefs	Home base directory to export across cluster with NFS for convenience of users.	masternode:/home
localfs	/usr/local directory to export across cluster with NFS.	masternode:/usr/local
pbshome	Value of PBS_HOME. See PBS documentation.	/var/spool/pbs
pbsprefix	Value of PBS_PREFIX. See PBS documentation.	/usr/local/pbs
pbsserver	Name of the node that is running the PBS server.	master

Parameter	Description	Possible values
scheduler	Name of the node that is running the Maui scheduler.	master
xcatprefix	Directory where xCAT files reside.	/opt/xcat
keyboard	Keyboard layout.	us
timezone	Name of the time zone used for the cluster.	US/Central
offutc	Time difference between UTC and local time (should be consistent with time zone parameter).	-6
mapperhost	Host that runs the GM mapper daemon.	masternode
serialmac	Console serial port number for MAC address collection; represents N in /dev/ttySN.	This value can be one of the following: '0' for COM A/1 '1' for COM B/2 'NA' for no MAC collection
serialbps	Baud rate across serial port.	9600
snmpc	SNMP community name. This is used by <b>mpasetup</b> .	Hardcoded to "public"
snmpd	The SNMP server address. This host will receive any alerts from the MPA cards. This is used by <b>mpasetup</b> .	10.0.0.1
poweralerts	Notify administrator by mail when nodes power on or off.	Y or N
timeservers	The source of the time synchronization signal.	masternode
logdays	Reserved for future use.	7
installdir	Directory containing all installation sources, Kickstart files, and post-installation files.	/install

Parameter	Description	Possible values
clustername	Reserved for future use.	wopr
dhcpver	Generate files for this version of DHCP.	2
dhcpconf	Full path of DHCP configuration file.	/etc/dhcpd.conf
dynamicr	Specify the dynamic range of IP addresses for a subnet. The parameters are: Interface, node type, DHCP server IP address, netmask, start and end of dynamic address range.	eth2,ia32,10.9.0.1,255.255.0.0,10.9.1.1,10.9.254.254
usernodes	List of all the nodes allowing direct user access.	user001,user002
usermaster	The user node that administers user accounts.	masternode
nisdomain	NIS domain if using NIS; NA if not available	NA yp.clusters.com
nismaster	Name of the master node if you use NIS; NA if not available.	NA nissserver.cluster.com
nisslaves	List of the slave nodes if you use NIS; NA if not available.	NA node065,node129
chagemin	Minimum number of days between password changes.	0
chagemax	Maximum number of days before a password expires.	60
chagewarn	The number of days of warning before a password change is required.	10
chageinactive	After a password has expired, this is the number of days of inactivity before the account is locked.	0

Parameter	Description	Possible values
mpcliroot	Location of the command line utility for accessing MPAs.	/opt/xcat/lib/mpcli
homelinks	Directory path to sym link entries for user home directories. Superceded by homefs.	NA
clusternet	No longer used.	
dynamic	No longer used.	
dynamictype	No longer used.	
clustervlan	No longer used.	
dynamiccb	No longer used.	



## nodelist.tab

This table describes all the nodes and other managed devices. It puts each of them in at least one group. Group names are selected by convention to aid in setting up the cluster and chosen based on the characteristics the devices share during setup or management operations.

nodelist table has one line per node. Example B-1 is the description of the nodelist.tab table.

*Example: B-1 Description of nodelist.tab table*

---

NodeName	GroupName,GroupName2,..,GroupNameN,Alias1,..,AliasN
----------	---

---

*Table B-3 Definition of nodelist.tab parameters*

Parameter	Description	Possible values
NodeName	This is the node or device name based on the naming convention used in the cluster. Any managed device should be included.	nodeX, where X is the node number. rsaNNN, ciscoNNN.. The only requirement is that the identifier has to begin with a non-numeric character.
GroupNameN	You can include the node in any and all groups. The groups are used to send a command to multiple nodes at the same time. For example, <b>rinv rack1</b> instead of <b>rinv node1,node2,node3.</b> Choosing group names and group membership wisely is a key to easier setup and management of a cluster.	The only requirement is that the identifier has to begin with a non-numeric character (that is, rack1, rack2, if the nodes are in different frames). You need a group "all" that includes all of the cluster's nodes.
AliasN	This is another name for the node. One use of an alias is to capture location information of the node.	posRRUU describes the physical location of the device. RR is the rack number and UU is the unit number for that device.

## noderes.tab

This table describes where the compute nodes can find the services they need.

Example B-2 describes where the group will find the external resources it needs for setup and access.

*Example: B-2 Description of noderes.tab table*

---

GroupName

---

*Table B-4 Definition of noderes.tab parameters*

Parameter	Description	Possible values
GroupName	Node or group name the parameters apply to.	The group or node should exist in the nodelist.tab table.
TFTP server	Name of the TFTP server to be used for the installation and boot processes.	Usually the host name of the master node unless multiple install servers are used.
NFS server	Name of the NFS server where the Red Hat CD-ROM and other install files reside.	Usually the host name of the master node unless multiple install servers are used.
NFS directory	Path name of the NFS exported Red Hat CD-ROM on NFS_Inst server.	By default, the installation directory is /install.
Serial	Console serial port number. Represents N in /dev/ttySN	This value can be one of the following: 0 for COM A/1 1 for COM B/2 NA for no serial console
Install Role	Indicates whether this node/group will be used as the installation node. This is only needed on large clusters where a small number of installation nodes are installed first and then used as install servers for remaining nodes (staging nodes).	The value can be only Y or N.

<b>Parameter</b>	<b>Description</b>	<b>Possible values</b>
BSD accounting	Turn on BSD accounting.	The value can be only Y or N.
Install GM software	Allow direct access to Myrinet for low-latency high-bandwidth IPC.	The value can be only Y or N.
Enable PBS	Install and configure PBS client.	The value can be only Y or N.
Access	access.conf support.	The value can be only Y or N.
GPFS	Install and configure GPFS client.	The value can be only Y or N.
Kickstart device	The network interface to be used for Kickstart.	Any network interface; for example, eth1.

## nodetype.tab

This table contains one line for each node and describes the name of the Kickstart file to use for its installation. With this table it is possible to have different nodes in the same cluster. For example, some nodes might not configure Gigabit Ethernet.

Example B-3 is the description of the nodetype.tab table.

*Example: B-3 Description of nodetype.tab table*

---

NodeName	KickstartFileName
----------	-------------------

---

*Table B-5 Definition of nodetype.tab parameters*

Parameter	Description	Possible values
NodeName	Name of the node this applies to.	nodeX, where X is the node number. The only requirement is that the identifier has to begin with a non-numeric character.
KickstartFileName	Name of the Kickstart template file used by xCAT to install this node.	The short name of a kstmpl file in /opt/xcat/ks73. By default use compute73.

## nodehm.tab

This table contains information on how to manage the hardware of each node. Because the devices used to install and manage a cluster could be different from one installation to another, this file allows xCAT commands to adapt themselves to your environment.

Example B-4 describes how to do some management functions for each node.

*Example: B-4 Description of nodehm.tab table*

---

```
NodeName    power,reset,cad,vitals,inv,cons,rvid,eventlogs,getmacs,netboot,  
eth0,gcons,serialbios
```

---

*Table B-6 Definition of nodehm.tab parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in "nodelist.tab" on page 193.	nodeX, where X is the node number.
Power	The method used by the <b>rpower</b> command to control the power to the node.	NA = The power function is not manageable.  rsa = The <b>rpower</b> command will use the RSA card to manage the node.  apc = The <b>rpower</b> command will use the APC MasterSwitch to manage the node.  apcp = The <b>rpower</b> command will use the APC MasterSwitch Plus to manage the node.

Parameter	Description	Possible values
Reset	The method used by the <b>rreset</b> command to do a hard reset on the node.	<p>NA = The reset function is not manageable.</p> <p>rsa = The <b>rreset</b> command will use the RSA card to manage the node.</p> <p>apc = The <b>rreset</b> command will use the APC MasterSwitch to manage the node.</p> <p>apcp = The <b>rreset</b> command will use the APC MasterSwitch Plus to manage the node.</p>
Cad (deprecated)	The method used by the <b>rcad</b> command to send Control+Alt+Delete (soft reset) to the node. This function does not work as intended and is deprecated.	<p>NA = The cad function is not manageable.</p> <p>asma= The rcad command will use the ASMA card to send the cad keys to a node.</p>
Vitals	The method used by the <b>rvitals</b> command to query the node for its vital sensors such as CPU, disk, and ambient temperature; voltages; fan speeds; etc.	<p>NA = The vitals function is not manageable.</p> <p>rsa= The <b>rvitals</b> command will use the RSA card to obtain the information from the node.</p>
Inv	The method used by the <b>rinv</b> command to query the node about its inventory data such as machine model, serial number, etc.	<p>NA = The inv function is not manageable.</p> <p>rsa = The <b>rinv</b> command will use the RSA card to obtain the information from the node.</p>

Parameter	Description	Possible values
Cons	The method used by the <b>rcons</b> and <b>wcons</b> commands to obtain a remote serial console for the node.	<p>NA = The cons function is not manageable.</p> <p>conserv = The commands will use conserv to obtain a console for the node.</p> <p>rtel = The commands will use reverse telnet to obtain a console for the node.</p> <p>tty = The commands will use a local tty to obtain a console from the node.</p>
EventLogs	The method used by the <b>reventlog</b> command to query and manage the management processor event log for the node.	<p>NA = The event log is not available.</p> <p>rsa = The <b>reventlog</b> command will use the RSA card to obtain and manage the event log.</p>
GetMacs	The method used by the <b>getmacs</b> command to obtain the MAC address of the node.	<p>NA = The function is not available; manual MAC collection only.</p> <p><b>rcons</b> = The <b>getmacs</b> command will use a remote console to obtain the MAC address of this node.</p> <p>cisco3500 = The <b>getmacs</b> command will query a Cisco 3500 switch to obtain the MAC address of this node.</p> <p>Unsupported options:</p> <p><b>extreme</b> = The <b>getmacs</b> command will query an Extreme network switch to obtain the MAC address of this node.</p>

Parameter	Description	Possible values
Netboot	The method used to boot the node over the remote network.	<p>NA = The node cannot boot over the network.</p> <p>pxe = For Pentium-based nodes that have PXE-enabled NICs.</p> <p>elilo = For Itanium-based nodes that have PXE-enabled NICs</p> <p>eb = For Pentium-based nodes that have Etherboot CD-ROMs or diskettes and cannot use PXE.</p>
eth0	The eth0 NIC type for this node. This is the NIC used for MAC address collection.	<p>NA = The node cannot boot over the network.</p> <p>eepro100 = Intel Pro/100+ based NIC.</p> <p>pcnet32 = AMD PC/Net32-based NIC.</p>



## mpa.tab

This table describes the Management Processor Adapters in this cluster and how to access their functions.

Example B-5 is the description of the mpa.tab table.

*Example: B-5 Description of mpa.tab table*

---

DeviceName      Type, InternalName, InternalNumber, Command, Reset, Rvid, Dhcp, Gateway

---

*Table B-7 Definition of mpa.tab parameters*

Parameter	Description	Possible values
DeviceName	This is the device name, taken from the list in "nodelist.tab" on page 193.	<i>deviceX</i> , where X is the device number. rsa001
Type	MPA type.	asma or rsa
InternalName	This identifier must be unique. This should be the DeviceName if the MPA is the primary management adapter for that device.	<i>deviceX</i> , rsa001
InternalNumber	This should be a unique ID number greater than 10000.	10001
Command	Access method.	telnet or mpcli
Reset	Access method.	HTTP (ASMA only), mpcli, NA
Rvid	Remote video method.	telnet (ASMA only), NA
Dhcp	DHCP-enabled.	Y or N (RSA only)
Gateway	Default gateway IP address.	10.0.0.1, NA

## apc.tab

This table gives xCAT the ability to power on or off any equipment connected to the APC MasterSwitch. It is intended for equipment that is not controlled by an MPA card (network switch, Equinox, Myrinet switch, etc.).

If your nodes do not use MPA hardware but they are connected to an APC MasterSwitch, then you still are able to control them through entries in this table.

Example B-6 is the description of the apc.tab table.

*Example: B-6 Description of apc.tab table*

---

EquipmentApcToUse, ApcPortNumber

---

*Table B-8 Definition of apc.tab parameters*

Parameter	Description	Possible values
Equipment	This is the name used with <b>rpower</b> or <b>rreset</b> commands.	This is the device name, taken from the list in "nodelist.tab" on page 193.
ApcToUse	This field is the APC alias, which has been chosen in the /etc/hosts table for this hardware.	Insert the alias name for the APC MasterSwitch. For example, apc1.
ApcPortNumber	This field is the port number where the device is plugged in.	Insert the port number that controls the device.

## apcp.tab

This table gives xCAT the ability to power on or off any equipment connected to the APC MasterSwitch Plus. It is intended for equipment that is not controlled by an MPA card (network switch, Equinox, Myrinet switch, etc.).

If your nodes do not use MPA hardware but they are connected to an APC MasterSwitch Plus, then you still are able to control them through entries in this table.

Example B-7 is the description of the apcp.tab table.

*Example: B-7 Description of apcp.tab table*

---

Equipment	ApcToUse, ApcUnitNumber, ApcPortNumber
-----------	--

---

*Table B-9 Definition of apcp.tab parameters*

Parameter	Description	Possible values
Equipment	This is the name used with <b>rpower</b> or <b>rreset</b> commands.	This is the device name, taken from the list in "nodelist.tab" on page 193.
ApcToUse	This field is the APC alias, which has been chosen in the /etc/hosts table for this hardware.	Insert the alias name for the APC MasterSwitch. For example, apc1.
ApcUnitNumber	This field is the unit number number where the device is plugged in.	Insert the APC unit number that controls the device, starting from the master, 1.
ApcPortNumber	This field is the port number where the device is plugged in.	Insert the port number on the master or slave unit that controls the device.

## mac.tab

This table will be generated by the getmacs script. It contains one line per node and describes the MAC address used for each node.

Do not edit this file yourself.

Example B-8 is the description of the mac.tab table.

*Example: B-8 Description of mac.tab table*

---

NodeName    MacAddressMgtCard

---

*Table B-10 Definition of mac.tab parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in "nodelist.tab" on page 193.	nodeX, where X is the node number.
MacAddressMgtCard	The MAC address of the management Ethernet adapter (eth0) in the node.	The card's MAC address.

## cisco3500.tab

This table maps the connections between the ports on the Cisco 3500 series switch and any cluster nodes using a switch, including storage nodes and user nodes.

Example B-9 is the description of the `apcp.cisco3500` table.

*Example: B-9 Description of `cisco3500.tab` table*

---

NodeName	CiscoToUse,CiscoPortNumber
----------	----------------------------

---

*Table B-11 Definition of `cisco3500.tab` parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in "nodelist.tab" on page 193.	nodeX, userX, or storageX, where X is the node number.
CiscoToUse	This field is the Cisco switch alias from "nodelist.tab" on page 193.	Insert the alias name for the Cisco switch. For example, cisco1.
CiscoPortNumber	This field is the port number where the node is plugged in.	Insert the port number for the network link.

## passwd.tab

This table is used during the installation process to access certain devices. The table is organized in a key-response format, as you might see with UNIX expect. The most common use within xCAT is to supply passwords for validating access to specific hardware (for example, a node, RSA, or ASMA card). If an xCAT script needs a password to log in to a specific device, it will look up that password in this table, indexed by the device identifier (DevKey), which it knows. It also has the root password that it needs to set on all nodes. It contains two fields: The identifier that xCAT scripts use as a key, and the string (usually a password) sent back to the device.

Example B-10 is the description of the password.tab table.

*Example: B-10 Description of passwd.tab table*

---

DevKey	Response
--------	----------

---

*Table B-12 Definition of passwd.tab parameters*

Parameter	Description	Possible values
DevKey	This field describes the device query that is requesting validation before allowing access. It is analogous to a database key or ID.	Values for this field are not defined because they depend on your hardware, but you need at least one entry for the connection to the node. These values are hardcoded in the xCAT scripts.  Possible values are: <ul style="list-style-type: none"><li>▶ For the connection to the node, you add the key rootpw.</li><li>▶ If you use ASMA and/or RSA hardware, you need to add two keys to this file:<ul style="list-style-type: none"><li>– asmauser</li><li>– asmapass</li></ul></li><li>▶ If you use Cisco products, use the key cisco.</li></ul>

Parameter	Description	Possible values
Response	This field contains the string used to respond to a device's validation (login or password) query.	<p>Again, the content of this field will depend on the value used to install the hardware.</p> <p>By default, for the ASMA/RSA hardware, the value for asmauser and asmapass are:</p> <ul style="list-style-type: none"> <li>▶ USERID</li> <li>▶ PASSWORD</li> </ul> <p>The value for the rootpw is the password you select during the Red Hat installation process.</p>

## conserver.tab

This table is used to map console servers to nodes. It is used only if `conserver` is being used to manage these serial ports.

Example B-11 is the description of the `conserver.tab` table.

*Example: B-11 Description of `conserver.tab` table*

---

NodeName	HostToUse, ConsoleName
----------	------------------------

---

*Table B-13 Definition of `conserver.tab` parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in “ <code>nodelist.tab</code> ” on page 193.	nodeX, userX, or storageX, where X is the node number.
HostToUse	This field contains the host name or IP address of the console server for this node.	masternode.
ConsoleName	This is the label used for this console. This identifier has to agree with the entry in <code>conserver.cf</code> .	node001-con.



## rtel.tab

This table is used to map serial ports to nodes when using a terminal server that uses a reverse telnet model (Equinox ELS or iTouch). It is used only if conserver is not being used to manage these serial ports.

Example B-12 is the description of the rtel.tab table.

*Example: B-12 Description of rtel.tab table*

---

NodeName	HostToUse,RtelPortNumber
----------	--------------------------

---

*Table B-14 Definition of rtel.tab parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in "nodelist.tab" on page 193.	nodeX, userX, or storageX, where X is the node number.
HostToUse	This field contains the host name or IP address of the terminal server for this node.	els001, itouch001.
RtelPortNumber	This is the IP port number used by this node. The IP port to physical port mapping is determined by the manufacturer.	3001, 2500.

## tty.tab

This table is used to map serial ports to nodes when using a terminal server that uses a tty model (Equinox ESP). It is used only if conserver is not being used to manage these serial ports.

The /dev files assigned to a specific ESP unit are a function of the sequential ESP number and can be determined by running the espcfg program in /etc/eqnx.

Example B-13 is the description of the tty.tab table.

*Example: B-13 Description of tty.tab table*

---

NodeName	HostToUse, TtyDevName
----------	-----------------------

---

*Table B-15 Definition of tty.tab parameters*

Parameter	Description	Possible values
NodeName	This is the node name, taken from the list in "nodelist.tab" on page 193.	nodeX, userX, or storageX, where X is the node number.
HostToUse	This field contains the host name or IP address of the terminal server for this node.	els001, itouch001.
TtyDevName	This is the device special file used to access the port assigned to this node.	/dev/ttyQ01e0.



# C

## Other hardware components

This appendix provides configuration details for hardware components frequently used in IBM Linux clusters, but not present in our lab configuration. xCAT provides a modular and flexible architecture that easily allows the incorporation of new hardware. This section provides information for these devices:

- ▶ IBM Advanced System Management Adapter (ASMA) Management Processor Adapters.
- ▶ Equinox ESP-8 and ESP-16 terminal servers
- ▶ iTouch Communications InReach 8000 series terminal servers
- ▶ Myricom Myrinet 2000 network

# IBM Advanced Systems Management Adapter

Setup for the ASMA card is very similar to the RSA adapter. The ASMA card uses the same architecture as the onboard service processor on the x330.

The ASMA card configuration utility is available on the xSeries 330 and IntelliStation R Pro - Advanced System Management Processor Firmware Update Utility (Version 1.07 at the time of this writing). This diskette image contains both the Service Processor firmware for the x330 node and the utility program used for configuring the onboard service processor and the ASMA card. The ASMA card can also be configured using the Advanced System Management PCI Adapter Firmware Update Diskette (Version 2.15 at the time of this writing). You may have obtained these diskettes to update firmware in the hardware preparation phase. Boot the node containing the ASMA card using either one of these diskettes.

At the configuration prompt, select **Configuration Settings -> Systems Management Adapter** and apply the following changes.

- ▶ Under General Settings:
  - a. Set the Systems Management Processor clock (time and date).
  - b. Toggle Set Clock to YES.
  - c. Press F6 to commit the changes.
- ▶ Under Network Settings:
  - a. Enable the network interface.
  - b. Set local IP address for the ASMA network interface.
  - c. Set the subnet mask.
  - d. Set the gateway.
  - e. Set the Data Rate to 10M or AUTO.
  - f. Set the Duplex to HALF or AUTO.
  - g. Press F6 to commit the changes.

The remainder of the setup can be handled by `mpasetup`, just as with the RSA card.

## Equinox ESP Terminal Servers

The ESP terminal server needs drivers installed on the host to provide remote serial ports. It is distributed by Equinox as a standard RPM package for 2.2 and 2.4 Linux kernels. The procedure to install the driver was obtained from Equinox documentation provided in the Equinox CD-ROM.

For more information see the Web site at:

<http://www.equinox.com/>

The ESP unit is installed as follows:

1. Record the MAC address from the small label on the rear of the unit. Keep this information with your cluster, or label it clearly on the front of the ESP where it can be easily read.
2. Attach an Ethernet cable to the local network.
3. Add the name and MAC address to the xCAT `mac.tab` file, and run `makedhcp` to refresh `/etc/dhcpd.conf` and the `dhcpd` daemon.
4. Power on the ESP. If the ESP has previously been programmed with an IP address, you may have to clear the configuration by holding down the reset button on the front panel for 10 seconds until a fast blink cycle of the LED's starts. The reset switch requires only a moderate pressure; do not press too hard or you will cause the switch to stick. Confirm through `/var/log/messages` that the DHCP request was received by the master node (Example C-1).

*Example: C-1 BOOTP requests from the ESP*

---

```
Jun 21 17:04:31 masternode dhcpd: BOOTREQUEST from 00:80:7d:80:d6:0e via eth1
Jun 21 17:04:31 masternode dhcpd: BOOTREPLY for 10.1.1.162 to esp001
(00:80:7d:80:d6:0e) via eth1
```

---

5. Wait a few minutes and verify that the ESP is correctly inserted on the network. Use the `ping` command to test communications (Example C-2).

*Example: C-2 Checking communications with the ESP*

---

```
[root]# ping -c3 esp001
PING esp001 (10.1.1.162) from 10.1.0.1: 56(84) bytes of data.
64 bytes from esp001 (10.1.1.162): icmp_seq=1 ttl=60 time=0.877ms
64 bytes from esp001 (10.1.1.162): icmp_seq=1 ttl=60 time=0.842ms
64 bytes from esp001 (10.1.1.162): icmp_seq=1 ttl=60 time=0.869ms

--- esp001 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss, time 2011ms
rtt mn/avg/max/mdev = 0.843/0.876/0.927/0.002 ms
```

---

Install the ESP driver:

1. Mount the media that contains the ESP driver, or download it from the Equinox Web site. You must have Version 3.03 or higher to support Red Hat 7.3 and the 2.4.x series of kernels.
2. Install using RPM (Example C-3 on page 214).

```
[root]# rpm -ivh esp*
```

*Example: C-3 Installing the espX driver rpm*

---

```
root@master /root]# cd /tmp
[root@master /tmp]# ls
espX-3.03-1.i386.rpm  install.log
[root@master /tmp]# rpm -ivh espX-3.03-1.i386.rpm
Preparing...          ##### [100%]
   1:espX              ##### [100%]
... rebuild and install device driver and utilities -- please wait...
... NOTICE: build log can be found at /usr/src/espX/buildlog
... create device files
... install man pages and update man page whatis
```

NOTE!!! - The configuration utility /etc/eqnx/espXcfg must be invoked to discover and configure ESPs before they can be used.

---

3. The espX package installation adds several lines to /etc/rc.local. This file is executed last in the initialization sequence, so the espX may not be available to services that require it earlier. Remove the lines flagged by # Equinox espX in /etc/rc.local and copy the initialization scripts provided by xCAT.

```
[root]# cp /opt/xcat/rc.d/espX /etc/rc.d/init.d/
[root]# chkconfig espX on
```

4. Run the espXcfg to initialize the ESP.

```
[root]# cd /etc/eqnx
[root]# ./espXcfg
```

You should see a configuration screen like the one shown in Example C-4.

**Note:** The espXcfg utility tries to search for ESPs on a subnet, and may not correctly identify the attached subnets or search all of the interfaces on the master node. If it does not find any units, you will need to install them manually.

*Example: C-4 ESP configuration screen*

---

```
+-----+
|Change ESP Configuration   v3.03|
+-----+
|Units:|
|ID           IP Address      ESP Number/Devices   ESP Model|
|00-80-7D-80-D6-0E  10.1.1.162      1 /dev/ttyQ01e0-01ef  10/100 ESP-16|
+-----+
```



6. Once the ESP has been found, you will get a screen like Example C-6. The ESP number is important: It determines which special device files will be associated with this ESP. You will need these values to configure your `conserver.cf` file.

*Example: C-6 ESP installation screen*

```
+-----+
|Install ESP: Specified ESP Found
|
|The following unit was found at the IP Address specified:
|
|    Unit ID: 00-80-7D-80-D6-0E   ESP Model: 10/100 ESP-16
|
|    IP Address: 10.1.1.162       ESP Number: 1
|
|This unit will be assigned the ESP number 1, as shown, making
|the following devices available for use: /dev/ttyQ01e0-01ef.
|You may Change the ESP number, or select Finish to install the
|unit on this host.
|
|
|
|
|    < Back   Finish   Change ESP number   Cancel   Help
|
+-----+
```

7. On the Change ESP configuration screen (Example C-4 on page 214), select **Exit**.

If your cluster uses the Equinox ESP terminal server you need to write a line for each node that looks like this:

```
nodename-con:/dev/ttyQXXxx:9600p:&:
```

When ESP terminal servers are utilized, a module is installed on the management node that creates a link between a virtual character device (`/dev/ttyQXXxx`) and the remote port on an ESP. The first ESP configured to be accessed by this module uses number 01 to designate itself, while the second will use 02, and so on. The ports on each of these uses a hex base numbering scheme with port 1 = e0 and port 16 = ef. This information can be retrieved by using the ESP configuration utility. Assuming node1 is attached to ESP 1 port 1, the `conserver.cf` entry would be:

```
node1:/dev/ttyQ01e0:9600p:&:
```



**Tip:** The `espcfg` creates the file `/etc/eqnx/esp.conf`.

```
[root]# more esp.conf
/dev/esp1 00-80-7D-80-5C-0A 192.168.1.155 4000 4000 0.0.0.0 255.255.255.0 16
10/100
```

It also creates `/dev/ttyQ(x)e(y)` files.

```
/dev/ttyQ1e0 to /dev/ttyQ1ef for the ESP number 1
/dev/ttyQ20e0 to /dev/ttyQ20ef for the ESP number 20
```

If you are installing multiple ESPs, and the units have been properly assigned IP addresses via DHCP, you may add additional ESPs by editing the `esp.conf` file and adding entries manually.

If the `expx` daemon needs to be restarted, use these steps:

```
[root]# service expx stop
[root]# rmmod expx
[root]# service expx start
```

The `expx` RPM compiles the `expx` driver against the currently installed kernel, so you must have the appropriate kernel code installed, and you must repeat this step any time you update the kernel on your management node. To rebuild the Equinox driver, remove the `expx` RPM using `rpm -e expx` and repeat the steps shown in Example C-3.

## iTouch Communications IR-8000 Terminal Servers

The In-Reach 8000 series of terminal servers has also been used successfully with xCAT, and a configuration script is provided. The IR-8000 series offers units with 4, 8, 20, and 40 ports, making it a useful option when the number of units in a rack is not a multiple of 16. As with other management equipment, the first step is to set up an IP address. This is done by attaching a serial cable to the local management port on the unit (this will be the highest numbered port). Connect this to a serial terminal, terminal emulator, or a serial port on your server node. Connect at 9600 baud, and set up the IP address following the example in Example C-7.

*Example: C-7 iTouch IP addressing dialog*

---

```
login>access

Enter username>root
iTouch>set priv
password>system
iTouch>define server ip address 10.1.1.163
```

```
iTouch>devine server ip subnet mask 255.255.0.0
iTouch>init server
```

---

This will reinitialize the terminal server and set the IP and subnet mask. From the master node, run:

```
[root]# setupitouch 10.1.1.163
```

This will set the ports to the correct mode for monitoring the serial consoles.

**Note:** You can also use DHCP to assign the IP, then run **setupitouch** command.

The setupitouch script is hardwired for the 20 port units, and configures all 20 ports for console access, removing the local management port. If you do not need 20 console ports, or you are using the 4, 8, or 40 port version, you may wish to modify the script to configure a different number of ports (for example, you might wish to configure ports 1–19 or ports 1–39 as console ports and leave the high numbered port as a local management port).

The iTouch uses a reverse telnet model like the ELS, so the configuration files will be similar except for port numbering. iTouch port numbers start at 2100 for the first port and increment by 100. The conserver.cf entry for a node attached to an iTouch would look like this:

```
node012:!itouch1:4200:&:
```

If you are not using conserver, the iTouch uses the rtel.tab file for configuration, and the entry looks like this:

```
node012:itouch1:4200
```

## Myrinet

Myrinet components provide cost-effective, low-latent, and high-speed interconnects for parallel applications. They can achieve theoretical bandwidths up to 2 Gbps and latencies as low as 7 microseconds. In order to achieve these metrics the Myrinet layout must be designed properly. A term frequently used with Myrinet layouts is *full bi-section bandwidth*.

If your cluster contains a Myricom Myrinet switch and a Myrinet card in a subset of the nodes, then you can use the instructions presented at “Setting up the Myrinet switch” on page 221 to set up the Myrinet network.

## Myrinet switch layout

In order to realize optimum performance of your Myrinet network there are a few design models that can be applied to switch layouts. It is recommended that if you are designing your own Myrinet cluster you should consult the Myrinet Web site at:

<http://www.myri.com>

### Single switch layout

Myrinet switches can provide up to 16 8-port blades within a single chassis. This means that for configurations up to 128 nodes they are quite simple, as shown in Figure C-1. Up to 16 active port blades (active blades denote that they are utilized to connect to Myrinet host adaptors) can provide this connectivity.

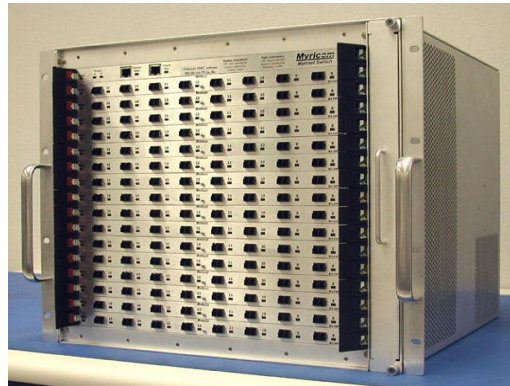


Figure C-1 Myrinet - Single switch layout

### Tree switch layout

When the number of nodes exceeds 128, multiple switches must be used. In order to provide full bi-section bandwidth, the appropriate number of paths and distribution to the entire cluster must exist. This is achieved with *spine* or passive switch blades (passive blades denote they are utilized to connect to passive blades on other switches). Figure C-2 on page 220 depicts an example 256 node cluster where spine switch ports are employed. Each switch in the top layer will have eight active blades of 64 ports connecting to the nodes, while the remaining eight spine or passive blades of 64 ports are divided evenly to the two switches in the spine layer. Both switches in the spine layer have all passive blades.

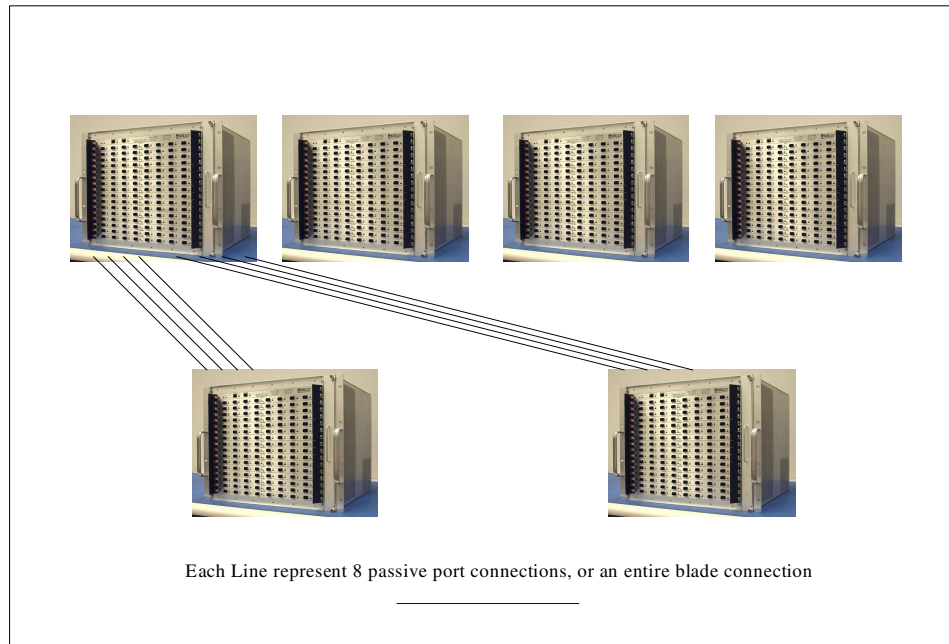


Figure C-2 Myrinet - Tree switch layout

### Polygon switch layout

Some cluster configurations built with numbers of nodes that are not powers of two can sometimes be more difficult to apply the tree design method to for cost and cluster design reasons. In some cases assumptions of node saturation capabilities can be applied to create layouts that may not provide full bi-section bandwidth, but are capable of sustaining the maximum throughput of the cluster. For example, a cluster with 384 nodes would require up to 10 switches utilizing the pyramid layout. With a polygon layout, as depicted in Figure C-3 on page 221, fewer switches are utilized while still providing optimum performance and latency. In the figure we see switch one connect spine blades 1–5 to the remaining switches and spine blades 6–8 to switches 2, 3, and 4. Switch two will connect spine blades 1–5 to the remaining switches and spine, and spine blades 6-8 to switches 3, 4, and 5. Each switch has eight active blades of 64 ports to provide node connects for 384 nodes.

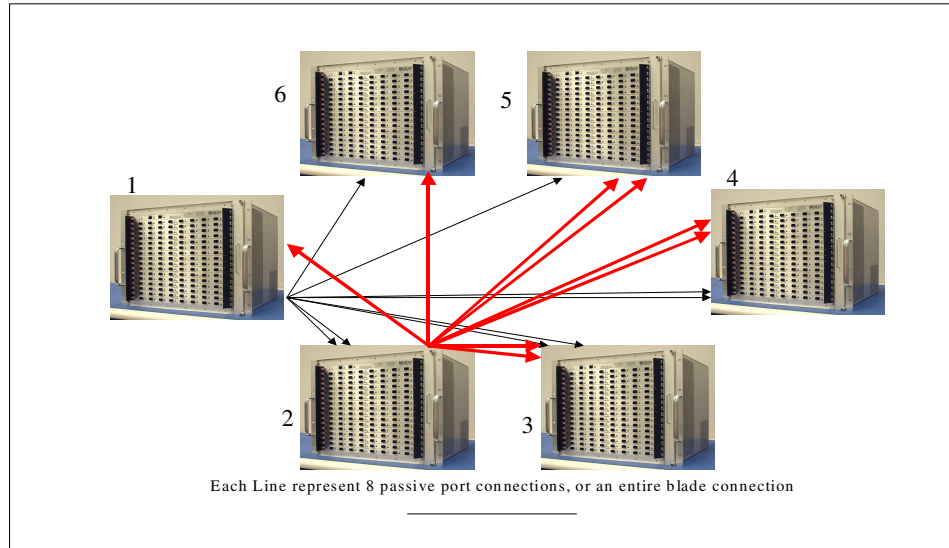


Figure C-3 Myrinet - Polygon switch layout

**Note:** Myrinet sSwitch design can become even more complex than the methods stated previously. We recommend that if you are designing your own Myrinet cluster you should consult the Myrinet Web site at:

<http://www.myri.com>

## Setting up the Myrinet switch

The Myrinet switch contains a monitoring line card with dual redundant Ethernet ports. This Ethernet interface allows access to the management capabilities of the Myrinet switch.

The monitor card can only be configured using BOOTP/DHCP. Record the MAC address that is printed on a label on the monitor card in `mac.tab`:

```
[root]# vi /opt/xcat/etc/mac.tab
myri001 00:60:DD:7F:35:29
```

Run `makedhcp` to configure the DHCP server.

```
[root]# makedhcp myrinet-hostname
```

Power on the Myrinet switch or, if it is already powered on, reset it by removing the power cord for a short time. Check your logs to see if the switch successfully

requested an IP address. Once the switch is configured, you can access it using a Web browser and the name you assigned to your Myrinet switch:

```
http://myri001
```

**Tip:** Mute is a graphical monitoring tool for Myrinet 2000 networks. We highly recommend that you install this after you install the Myrinet software. Mute allows you to graphically display the connections in the Myrinet network and it will highlight any problematic cables and components you may have. Without this, locating and fixing problems on the Myrinet network is very difficult.

The instructions for downloading and installing Mute can be found on Myricom's Web site:

```
http://www.myri.com/scs/mute/
```

## Installing the Myrinet software

The software for Myrinet is called GM, which stands for Glenn's Messages (<http://www.myri.com/scs/faq/faq-gm.html#gmq1>). GM can be downloaded from the Myricom Web site at <http://www.myri.com/> but you will need to apply for a user name and password.

**Note:** At the time of this publication, the latest tested version of GM was 1.5.2.1.

xCAT contains a script to build an RPM package for the GM code. We build the code in the /tmp directory.

```
[root]# cp gm-1.5.1_Linux.tar.gz /tmp
[root]# cd /tmp
[root]# /opt/xcat/build/gm/gmmaker 1.5.1_Linux
```

The result will be an RPM package in /usr/src/redhat/RPMS/i686. The name of the RPM package depends on the kernel version. If the build fails, refer to the documentation and FAQ on the Myricom Web site.

The GM package needs to be installed on the management node if you want to use any of the management functions. The management node does not usually contain a Myrinet interface, so we prevent GM from starting at boot:

```
[root]# rpm -ivh /usr/src/redhat/RPMS/i686/gm-1.5.1_Linux-2.4.18-4smp.i686.rpm
[root]# chkconfig --del gm
```

The GM package will be installed on all the nodes, so it needs to be copied to the post-install directory:

```
[root]# cp /usr/src/redhat/RPMS/i686/gm-1.5.1_Linux-2.4.18-4smp.i686.rpm \
/install/post/kernel/
```

xCAT will only install the GM package on nodes that have a Y in the GM field in `noderes.tab`. In our example, all the nodes have a Myrinet card, so make sure that the GM field is set to Y.

```
[root]# vi /opt/xcat/etc/noderes.tab
compute  masternode,masternode,/install,0,N,N,N,Y,N,N,N,NA
storage  masternode,masternode,/install,0,N,N,N,Y,N,N,N,NA
```

**Tip:** If your cluster has a Myrinet card in a subset of the nodes, make sure that you have defined a set of groups that includes all the nodes that do not contain a Myrinet card.

For example, consider a cluster that has user nodes and compute nodes that do not contain Myrinet cards. It also has enhanced compute nodes and storage nodes with Myrinet cards. In this example you define a group called `user` to contain all the user nodes and a group called `scn`, which stands for standard compute node. On `noderes.tab`, these groups would be listed with a N in the GM field. The enhanced compute nodes are in the `ecn` group and the storage nodes are in the `storage` group. Both these groups have a Y in the GM field.

The post-installation script will install the version of GM indicated by the `GMVER` variable in the Kickstart file. Make sure that `compute73.kstmp` and `storage73.kstmp` contain the following line. Remember to rerun `mkks` when you modify the Kickstart templates.

```
[root]# vi /opt/xcat/ks73/compute73.kstmp /opt/xcat/ks73/storage73.kstmp
GMVER=1.5.1_Linux
```

The Myrinet card can be used with the TCP/IP protocol. The post-install script will query the DNS server to check if you have defined IP addresses for all the Myrinet interfaces. The host name should be `nodename-myri0`. It then uses the IP address you defined to configure the Myrinet interface. Check that you have defined the IP addresses for all the Myrinet nodes (Example C-8).

#### *Example: C-8 Checking Myrinet IP addresses*

```
[root]# for node in $(nr myri); do host ${node}-myri0; done
node001-myri0.clusters.com has address 10.2.1.1
node002-myri0.clusters.com has address 10.2.1.2
node003-myri0.clusters.com has address 10.2.1.3
node004-myri0.clusters.com has address 10.2.1.4
```

```
node005-myri0.clusters.com has address 10.2.1.5
node006-myri0.clusters.com has address 10.2.1.6
node007-myri0.clusters.com has address 10.2.1.7
node008-myri0.clusters.com has address 10.2.1.8
storage001-myri0.clusters.com has address 10.2.1.141
```

---

Now, you are ready to install the Myrinet software on the Myrinet nodes. If you are in the process of configuring the management node or installing the cluster, make sure you complete that process first and then continue here. If the cluster is already installed, you will need to reinstall the Myrinet nodes using **rinstall**.

```
[root]# rinstall myri
```

During the installation, keep an eye on the Myrinet switch. When the GM driver loads and all the Myrinet cables are correctly installed, the green LED on every port with a node connected to it should come on. If none of the lights come on, check for messages from the post-install routine in `/var/log/messages`.

Myrinet is a source-routed network. This means that every node must know the route to all other hosts. We need to run the GM mapper to create the routing information for each host and verify the connectivity between all the Myrinet nodes:

```
[root]# makegmroutes myri
[root]# psh myri /opt/xcat/sbin/gmroutecheck myri
```

The **gmroutecheck** command has no output when everything is correct. When there are any errors, we suggest that you install and use the Mute utility to diagnose any problems there may be. More information to help you find and fix possible problems can be found in the Myricom FAQ at:

[http://www.myri.com/scs/GM\\_FAQ.html](http://www.myri.com/scs/GM_FAQ.html)





# D

## Application examples

Configuring and building a high-performance computing cluster is quite interesting in itself, but like any other computer, it is only a tool and is useless without an application to run on it. This appendix provides information on how to install and run some basic parallel computing examples to help you test your cluster. These examples use publicly available code and can also be used as a starting point for developing your own parallel applications.

In this appendix we will discuss:

- ▶ Basic setup of user accounts
- ▶ MPICH
- ▶ POVray and MPI-POVray
- ▶ The HPL Linpack benchmark

## User accounts

Good practice dictates that you avoid running application code from the root account, as the root account will allow you to do terrible damage to your system by accident. The root account should be used only for administrative tasks that require it. There are numerous methods of managing user accounts on distributed systems, and a discussion of these is outside the scope of this document. For these examples we have used simple user management within the cluster nodes only.

First we will set up a user account on the master node, using the xCAT utility `addclusteruser`, as shown in Example D-1.

*Example: D-1 Adding a cluster user account*

---

```
[root]# groupadd ibm
[root]# addclusteruser
Enter username: ibm
Enter group: ibm
Enter UID (return for next): 501
Enter absolute home directory root: /home
Enter passwd (blank for random): red!b00k
Changing password for user ibm.
passwd: all authentication tokens updated successfully.
[root]# prsync /etc/passwd /etc/shadow /etc/group /etc/gshadow compute:/etc
[root]# gensshkeys ibm
[root]#
```

---

**Note:** You do not need to run `prsync` if NIS is being used. If you are not using NIS, you only need to copy (by using `prsync`) password and group information; the rest is a security risk.

You do not need to run `gensshkeys` if `ssh` is being used. The `addclusteruser` does it for you.

In these examples, commands run from the root account are shown with the # prompt, while commands run from a user account are shown with a \$ prompt.

## MPICH

MPICH is a freely available, portable implementation of MPI, the Standard for message-passing libraries. MPICH (often pronounced *em-pitch*) is maintained by Argonne National Labs (ANL) and Mississippi State University (MSU). The

application examples shown in this appendix use MPICH to do parallel computation.

MPICH must be specifically configured for each combination of cluster interconnect, compiler, and system architecture. For generality, we have provided here instructions for building MPICH for Ethernet and GNU compilers. xCAT's mpimaker utility provides facilities for configuring multiple versions of MPICH.

1. Learn about MPICH. MPICH's home page is at:

<http://www-unix.mcs.anl.gov/mpi/mpich/>

2. Download MPICH:

<ftp://ftp.mcs.anl.gov/pub/mpi/mpich.tar.gz>

3. Build MPICH:

```
> cd /opt/xcat/build/mpi
> cp download_directory/mpich.tar.gz .
> mv mpich.tar.gz mpich-1.2.4.tar.gz (or whatever the current version
acutally is)
> ./mpimaker (This will show you the possible arguments. You may want to
use different ones.)
> ./mpimaker 1.2.4 smp gnu ssh
```

You can use the command `tail -f mpich-1.2.4/make.log` to view the progress of the build.

When done, you should have code in `/usr/local/mpich/1.2.4/ip/smp/gnu/ssh`.

4. Adjust your environment. Add the following to `~/.bashrc`. You could also put it in `/etc/profile` if all cluster users will use only this MPI lib.

```
export MPICH="/usr/local/mpich/1.2.4/ip/smp/gnu/ssh"
export MPICH_PATH="${MPICH}/bin"
export MPICH_LIB="${MPICH}/lib"
export PATH="${MPICH_PATH}:${PATH}"
export LD_LIBRARY_PATH="${MPICH_LIB}:${LD_LIBRARY_PATH}"
```

5. Test the environment. After re-sourcing the environment changes that you have made, it is a good idea to validate that everything is correct. A simple, but not complete, way to do this is:

```
> which mpicc
```

If you are setup for MPICH as in the above example, the output of this command should be:

```
/usr/local/mpich/1.2.4/ip/smp/gnu/ssh/bin/mpicc
```

## Persistence of Vision Raytracer (POVray)

The increased computational power of High-Performance Computing has been exploited in many fields of biomedical, manufacturing, and research. One area in which HPC has also been consistently placed in action is the field of graphic rendering. POVray is a scripting language that describes a 3D environment, which is then parsed through an engine that translates the language into an image.

Applying POVray over a HPC environment allows what was once a serial operation to be executed in parallel. Image rendering (being generally a highly parallel process) can easily be split into smaller portions that can be distributed as smaller parts across compute nodes in a cluster.

POVray as distributed by its author is not a parallel application, but patches have been created to exploit the power of a parallel HPC environment by using MPI. The runs from POVray should not be used for benchmarks, because it is not optimized. Rather, POVray should be run as a demo that can visually verify in a customer environment that the HPC cluster is functioning and that the MPI libraries are correctly used. POVray also provides an excellent graphical demonstration of the application speedups possible through parallel computing.

### Serial POVray

The procedure below describes how to install POVray for demonstration usage and how to run it. All installation procedures should be done as root, but the execution of the compute job does not require root permissions.

**Note:** The POVray package and the MPI patch for POVray can be found at:

<http://www.povray.org>

Others POVray Web sites are:

<http://www.verrall.demon.co.uk/mpipov/> (MPI POVray)

<http://www-mddsp.enel.ucalgary.ca/People/adilger/povray/pvmpov.html>  
(PVM POVray)

Obtain the POVray package and the MPI patch for POVray. This procedure has been tested with:

- ▶ POVray Version 3.1g
- ▶ MPICH Version 1.2.2.3 and 1.2.4
- ▶ MPI-POV Patch Level 1.0

This procedure assumes an Intel Pentium architecture, and uses the precompiled POVray binaries for the single-CPU portion. To use this procedure on a different architecture you may need to download the POVray source and recompile it.

1. Install POVray on the management machine. The install script provided with the package moves the executable to `/usr/local/bin`, and the rest of the POVray package to `/usr/local/lib`.

```
[root]# cd /tmp
[root]# tar zxvf povlinux.tgz
[root]# cd povray31
[root]# ./install
```

2. Copy the POVray INI files and the scenes files into a user directory.

```
[root]# cd /usr/local/lib/povray31
[root]# cp povray.ini $HOME/.povrayrc
[root]# cp -R scenes $HOME
```

3. Insure that the POVray binaries are in a directory in the PATH environment variable. Adjust your \$PATH environment if necessary.

```
[root]# which x-povray
/usr/local/bin/x-povray
```

4. Add the following into the `.profile` file for the users who will use POVray. Alternately, you can add this to `/etc/profile` for all users.

```
export POVPATH=/usr/local/lib/povray31
export POVINI=${HOME}/.povrayrc
export POVINC=${POVPATH}/include
```

Add the environment variables to the current environment or reboot before starting POVray.

5. Make sure that the environment of the shell is updated before starting POVray. Depending on where you have set the environment variables, “source” your `.profile` file or the `/etc/profile` file, or reboot the machine.

```
[root]# . ~/.profile
[root]# printenv | grep POV
POVINC=/usr/local/lib/povray31/include
POVPATH=/usr/local/lib/povray31
POVINI=/home/sdenham/.povrayrc
```

6. Do a trial run of serial POVray running only on the management node.

```
[root]# cd $HOME/scenes/advanced/
[root]# x-povray -Iskyvase.pov -L${POVINC} -w640 -H480 +D +P
```

**Note:** x-povray has many command line options. They are described in the man page, or displayed by entering x-povray without operands. In our example:

- ▶ -I specifies the input (scene) file.
- ▶ -L specifies the location of the object library.
- ▶ -w specifies the image width.
- ▶ -H specifies the image height.
- ▶ +D tells POVray to display while rendering.
- ▶ +P tells POVray to pause the window at the end of the display.

At this point, POVray should be running on the management node. Ensure that you can get to this point without errors before continuing.

## Distributed POVray using MPI-POVray

Now that POVray has been installed onto the management node, we can install the MPI parallel version of POVray. The MPI version of POVray is not distributed in binary, so it has to be recompiled from source and a patch file.

1. Download the latest version of the POVray source and MPI-POVray patch from <http://www.verrall.demon.co.uk/mpipov/>. The files needed are:

- povuni\_s.tgz
- mpi-povray-1.0.patch.gz

2. Untar the files, placing them in the correct directories.

```
[root]# cd /usr/local
[root]# tar zxvf /tmp/povuni_s.tgz
[root]# cp /tmp/mpi-povray-1.0.patch.gz /usr/local/povray31
[root]# cd /usr/local/povray31
[root]# gunzip mpi-povray-1.0.patch.gz
```

3. Patch the POVray source with the MPI-POVray patch.

```
[root]# cat mpi-povray-1.0.patch | patch -p1
[root]# cd /usr/local/povray31/source/mpi-unix
```

4. Make sure that MPICH was configured to use `ssh`.
5. Make sure that `mpicc` points to the right version of MPICH, the one using `ssh`.
6. At this point check that the `PATH` of `mpicc` is located in your `PATH`. This should have already been done during the MPICH installation.

```
[root]# which mpicc
/usr/local/mpich/1.2.4/ip/smp/gnu/ssh/bin/mpicc
```

**Note:** If being used on Itanium-based computers, remove the `-m386` flag on the `FLAGS` on your compile options. This does not provide optimized performance, but it allows MPI-POVray to compile.

7. Compile the `mpi-povray` binary by executing `make`.

```
[root]# make
```

8. Add the remote access method for the compute nodes by adding the following to the local user's `.profile` file or `/etc/profile`.

```
export RSHCOMMAND="ssh"
```

9. Make sure that the environment of the shell is updated by either sourcing the `/etc/profile` file or `~/.profile` file, or rebooting the machine.

10. Move the compiled binary into a path that is available through `PATH`.

```
[root]# mv /usr/local/povray31/source/mpi-unix/mpi-x-povray /usr/local/bin
```

At this point you may delete the source tree from `povray31` if you wish.

11. You may wish to edit `/usr/local/mpich/share/machines.LINUX` to include all the nodes in the cluster to be used for computing. Examples of the formats of this file are given in Example D-2 and Example D-3. Alternatively, you may create your own private file, and reference it using the `-machinefile` keyword of the `mpirun` command. This is useful if you wish to try different combinations of nodes.

*Example: D-2 Newer format machines file for SMP systems*

---

```
# machines.LINUX file type 1
# using the syntax "hostname:proc"
node001:2
node002:2
node003:2
node004:2
node005:2
node006:2
node007:2
node008:2
```

---

*Example: D-3 Classical format machines file for SMP systems*

---

```
# machines.LINUX file type 2
# using the syntax of one processor per line
node1
node1
node2
node2
node3
```

```
node3
node4
node4
node5
node5
node6
node6
node7
node7
node8
node8
```

---

12. Log in to one of your nodes, and try executing the MPI-POVray binary on the chess2.pov file with the commands shown in Example D-4.

*Example: D-4 Parallel POVray on our cluster*

---

```
[ibm@masternode ibm] # ssh node001
Last login: Thu Jun 27 13:30:42 2002 from masternode
[ibm@node001 ibm] # . ~/.profile
[ibm@node001 ibm] # cd $HOME/scenes/advanced
[ibm@node001 ibm] # export DISPLAY=masternode:0.0
[ibm@node001 ibm] # mpirun -np 8 /usr/local/bin/mpi-x-povray \
-Ichess2.pov +L${POVINC} -w640 -H480 +D
```

---

At this point you should be able to see the output of POVray drawn on the screen. If it does not work, check that the user exists on all machines with the same password and does not require any interaction for login. Test this by accessing the nodes via secure shell (**ssh**).

If you are required to execute POVray without any display, remove the +D flag. If you wish to leave the display on the screen for a moment after it is rendered, add the +P flag.

**Note:** The management node processors can be included as part of the computation if they are included in the machines.LINUX file. However, we recommend that the number of processors used not be more than the total number of processors in the compute nodes. Increasing the number of processes to greater than the number of processors results in higher overhead and longer run time.

## High Performance Linpack (HPL)

HPL is a software package that solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers. It can thus be



regarded as a portable, as well as freely available, implementation of the High Performance Computing Linpack Benchmark. HPL was supported in part by a grant from the U.S. Department of Energy's Lawrence Livermore National Laboratory and Los Alamos National Laboratory as part of the ASCI project.

The HPL software package requires the availability on your system of an MPI library (we used MPICH), and either the Basic Linear Algebra Subprograms (BLAS) or the Vector Signal Image Processing Library (VISPL). We used the Automatically Tuned Linear Algebra Software (ATLAS) package from Sourceforge, which is available pre-compiled for numerous architectures.

## Installing ATLAS

To install ATLAS:

1. Download the appropriate ATLAS distribution from <http://math-atlas.sourceforge.net>. For our lab cluster we used `atlas3.4.1_Linux_PIIISSE1.tar.gz`.
2. Untar the file and copy the files `libatlas.a`, `libf77blas.a`, `libcblas.a`, and `liblapack.a` to `/usr/local/lib`. Copy the files `cblas.h` and `clapack.h` to `/usr/local/include`.

## Installing HPL

To install HPL:

1. Download the HPL distribution from:  
<http://www.netlib.org/benchmark/hpl>
2. Untar the distribution into a working directory. The distribution file `hpl.tgz` will expand into an `hpl` directory.
3. In the `hpl` directory, select the best match for your architecture from the subdirectory `hpl/setup`, and copy it to the parent directory. We chose the Linux Pentium II SSE1 implementation using the Fortran BLAS library.

```
[root]# cd ~hpl
[root]# cp setup/Make.Linux_PII_FBLAS ..
[root]# vi Linux_PII_FBLAS
```

4. Adjust the make file you just copied to point to the correct compilers and libraries. We left the default `gcc` and `g77` compilers, and changed these lines:

```
MPdir    = /usr/local/mpich/1.2.4/ip/smp/gnu/ssh
Adir     = /usr/local/lib
```

5. Compile the HPL application suite by using the **make** command and specifying the architecture you have selected.:

```
make arch=Linux_PII_FBLAS
```

6. Create a host file in your home directory containing the names of the nodes you wish to run hpl on:

```
[root]# echo -e "node001\nnode002\nnode003\nnode004\n" > ~/hosts
```

7. Execute the HPL sample run from the architecture-specific binary file directory.

```
[root]# cd ~/hpl/bin/Linux_PII_FBLAS
[root]# mpirun -np 4 -machinefile ~/hosts xhpl
```

The xhpl application will run on your selected nodes and produce output as in Example D-5.

*Example: D-5 Output*

---

```
=====
HPLinpack 1.0 -- High-Performance Linpack benchmark -- September 27, 2000
Written by A. Petitot and R. Clint Whaley, Innovative Computing Labs., UTK
=====
```

An explanation of the input/output parameters follows:

T/V : Wall time / encoded variant.  
N : The order of the coefficient matrix A.  
NB : The partitioning blocking factor.  
P : The number of process rows.  
Q : The number of process columns.  
Time : Time in seconds to solve the linear system.  
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

```
N      :      29      30      34      35
NB     :      1      2      3      4
P      :      2      1      4
Q      :      2      4      1
PFACT  : Left    Crout   Right
NBMIN  :      2      4
NDIV   :      2
RFACT  : Left    Crout   Right
BCAST  : 1ring
DEPTH  :      0
SWAP   : Mix (threshold = 64)
L1     : transposed form
U      : transposed form
EQUIL  : yes
```

ALIGN : 8 double precision words

- 
- The matrix A is randomly generated for each test.
  - The following scaled residual checks will be computed:
    - 1)  $\frac{\|Ax-b\|_{\infty}}{\text{eps} * \|A\|_1 * N}$
    - 2)  $\frac{\|Ax-b\|_{\infty}}{\text{eps} * \|A\|_1 * \|x\|_1}$
    - 3)  $\frac{\|Ax-b\|_{\infty}}{\text{eps} * \|A\|_{\infty} * \|x\|_{\infty}}$
  - The relative machine precision (eps) is taken to be 1.110223e-16
  - Computational tests pass if scaled residuals are less than 16.0

=====

T/V	N	NB	P	Q	Time	Gflops
W00L2L2	29	1	2	2	0.08	2.102e-04

-----

$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _1 * N}$	=	0.0674622	.....	PASSED
$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _1 * \ x\ _1}$	=	0.0519667	.....	PASSED
$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _{\infty} * \ x\ _{\infty}}$	=	0.0174238	.....	PASSED

=====

...

$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _1 * N}$	=	0.0313155	.....	PASSED
$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _1 * \ x\ _1}$	=	0.0343347	.....	PASSED
$\frac{\ Ax-b\ _{\infty}}{\text{eps} * \ A\ _{\infty} * \ x\ _{\infty}}$	=	0.0120026	.....	PASSED

=====

Finished 864 tests with the following results:  
864 tests completed and passed residual checks,  
0 tests completed and failed residual checks,  
0 tests skipped because of illegal input values.

-----

End of Tests.

=====

---



# Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

## IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 240.

- ▶ *Linux Clustering with CSM and GPFS*, SG24-6601
- ▶ *Linux HPC Cluster Installation*, SG24-6041

## Other resources

These publications are also relevant as further information sources:

- ▶ *IBM Cluster Systems Management for Linux: Administration Guide*, SA22-7873
- ▶ *IBM Cluster Systems Management for Linux: Hardware Planning and Control Guide*, SA22-7856
- ▶ *IBM Cluster Systems Management for Linux: Planning and Installation Guide*, SA22-7853
- ▶ *IBM Cluster Systems Management for Linux: Technical Reference*, SA22-7851

## Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Advanced TFTP (ATFTP) download  
<ftp://ftp.mamalinux.com/pub/atftp/atftp-0.3.tar.gz>
- ▶ APC switch download  
<http://www.apc.com/>
- ▶ Conserver Web site  
<http://www.conserver.com>

- ▶ CSM information  
<http://www.ibm.com/servers/eserver/clusters/library/>
- ▶ Equinox Systems Inc. Web site  
<http://www.equinox.com/>
- ▶ Glenn's Messages (GM) software for Myrinet  
<http://www.myri.com/scs/faq/faq-gm.html#gmq1>
- ▶ GNU Project Web site  
<http://www.gnu.org/>
- ▶ High-Performance Computing Information Web site  
<http://www.top500.org/>
- ▶ HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers  
<http://www.netlib.org/benchmark/hpl>
- ▶ IBM developerWorks Web site  
<http://www.ibm.com/developerworks/>
- ▶ IBM @server Cluster Web site  
<http://www.ibm.com/servers/eserver/clusters/>
- ▶ IBM PC Support Web Site  
<http://www.pc.ibm.com/support/>
- ▶ IBM Storage Solutions Web site  
<http://www.ibm.com/storage/>
- ▶ Intel PRO/100 FastEthernet Adapter  
<http://support.intel.com/support/network/>
- ▶ iTouch Communications/MRV Communications Web sites  
<http://www.itouchcom.com/>  
<http://www.mrv.com/>
- ▶ iTouch Terminal server information  
[http://www.mrv.com/products/remote\\_management/products](http://www.mrv.com/products/remote_management/products)
- ▶ Linux at IBM Web site  
<http://www.ibm.com/linux/>
- ▶ Linux Documentation Project Web site  
<http://www.linuxdoc.org>

- ▶ Linux kernel archives  
<http://www.kernel.org/>
- ▶ Linux Kernel Crash Dump Project Web site  
<http://lkcd.sourceforge.net/>
- ▶ Matt Bohnsack page  
<http://bohnsack.com>
- ▶ Mirror sites  
<http://www.redhat.com/download/mirror.html>
- ▶ MPICH homepage  
<http://www-unix.mcs.anl.gov/mpi/mpich>
- ▶ MPI-POVray  
<http://www.verrall.demon.co.uk/mpipov>
- ▶ Mute download and installation instructions  
<http://www.myri.com/scs/mute/>
- ▶ Myricom FAQ  
[http://www.myri.com/scs/GM\\_FAQ.html](http://www.myri.com/scs/GM_FAQ.html)
- ▶ Myricom, Inc. Web site  
<http://www.myri.com/>
- ▶ Network Time Protocol Web site  
<http://www.ntp.org/>
- ▶ NTP service and software  
<http://www.eecis.udel.edu/~ntp/>  
<http://www.ntp.org/>
- ▶ OpenSSH Project Web site  
<http://www.openssh.org/>
- ▶ Perl Web site  
<http://www.perl.com/>
- ▶ Pre-built custom kernels for versions of Red Hat lower than 7.3  
<http://x-cat.org/download/xcat/>
- ▶ POVray Project  
<http://www.povray.org>
- ▶ PVM patch for POVRay  
<http://www-mddsp.enel.ucalgary.ca/People/adilger/povray/pvmpov.html>

- ▶ Red Hat 7.3 installation details  
<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/install-guide/>
- ▶ Red Hat installation updates  
<http://www.redhat.com/support/errata>
- ▶ Red Hat updated packages  
<ftp://updates.redhat.com/7.3/en/os/>
- ▶ Red Hat Web site  
<http://www.redhat.com/>
- ▶ Latest Intel e1000 driver download  
<http://support.intel.com/support/network/adapter/1000/software.htm>
- ▶ Supercluster.org Web site  
<http://www.supercluster.org>
- ▶ SYSLinux Project  
<http://syslinux.zytor.com/>
- ▶ The ATLAS (Automatically Tuned Linear Algebra Software) Project  
<http://math-atlas.sourceforge.net>  
<ftp://ftp.mcs.anl.gov/pub/mpi/mpich.tar.gz>
- ▶ xCAT download and application for a user ID and password to get download  
<http://x-cat.org/download/xcat/>
- ▶ xCAT man pages  
<http://x-cat.org/docs/man-pages/>
- ▶ xCAT Project  
<http://x-cat.org>
- ▶ xCAT Remote Flash information  
<http://x-cat.org/docs/flash-HOWTO.html>
- ▶ xServer Models x345 and x335 information  
<http://www.pc.ibm.com/us/eserver/xseries/>

## How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)



You can also download additional materials (code samples or diskette/CD-ROM images) from that site.

## **IBM Redbooks collections**

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.



# Glossary

**ASMA** Advanced Systems Management Adapter. Remote supervisor adapter. This is an older IBM MPA card.

**Beowulf** An approach to building a cluster of off-the-shelf commodity personal computers, interconnecting them with Ethernet, and running programs written for parallel processing.

**Bootstrap Protocol (BOOTP)** A protocol that allows a client to find both its Internet Protocol (IP) address and the name of a file from a network server.

**Cluster** A loosely-coupled collection of independent systems (nodes) organized into a network for the purpose of sharing resources and communicating with each other.

**Domain Name System (DNS)** A server program that supplies name-to-address conversion by mapping domain names to IP addresses. The domain name server allows users to request services of a computer by using a symbolic name, which is easier to remember than an IP address.

## **Dynamic Host Configuration Protocol**

**(DHCP)** An application-layer protocol that allows a machine on the network, the client, to get an IP address and other configuration parameters from the server.

**ELS** Equinox terminal server. This device allows you to connect many RS-232 serial devices (especially terminals) through a single LAN connection.

**ESP** Ethernet Serial Provider. This Equinox device family allows you to access multiple serial RS-232 serial devices through a single LAN connection.

**High Performance Computing cluster (HPC cluster)** A system designed for greater computational power than a single computer alone could provide. In the past, high-performance computing was typically reserved for the scientific community, but now it is breaking into the mainstream market. HPC is a field of computer science that focuses on developing supercomputers, parallel processing algorithms, and applications.

**Inter-Process Communication (IPC)** The process by which programs communicate data to each other and synchronize their activities. Semaphores, signals, and internal message queues are common methods of inter process communication.

**Kickstart** Allows users to automate most of a Red Hat Linux installation. A system administrator can create a single file containing the answers to all the questions that would normally be asked during a typical Red Hat Linux installation.

## **Message Passing Interface (MPI)**

Message-passing standard that facilitates the development of parallel applications and libraries.

**MPA** Management Processor Adapter. A gateway between a proprietary management processor network (MPN) and an external protocol (Ethernet or RS-232). ASMA and RSA cards are examples of MPA cards.

**MPN** Management processor network. This network connects an MPA and a chain of onboard systems management processors (typically eight) together in a daisy chain.

**Network File System (NFS)** A distributed file system that allows users to access files and directories located on remote computers and to treat those files and directories as if they were local.

**Open source** Software that is developed and improved by a group of volunteers cooperating together on a network. Many parts of the UNIX-like operating system were developed this way, including Linux.

**Preboot Execution Environment (PXE)**

Designed to define a standard set of preboot protocol services within a client, with the goal of allowing network-based booting to use industry-standard protocols.

**PXELinux** A SYSLinux derivative for booting Linux off a network server using a network ROM conforming to PXE specifications.

**Red Hat Package Manager (RPM)** A packaging system that allows users to package source code for new software into source and binary form such that binaries can be easily installed and tracked and source can be rebuilt easily.

**Redundant Array of Independent Disks**

**(RAID)** A set of physical disks that act as a single physical volume and use parity checking to protect against disk failure.

**RSA** Remote Supervisor Adapter allows remote management similar to ASMA. However, RSA provides additional functions over the ASMA card. RSA is used with the new models like the Model 342.

**Simple Network Management Protocol**

**(SNMP)** A protocol governing network management and the monitoring of network devices and their functions.

**Small computer system interface (SCSI)** An adapter supporting the attachment of various direct-access storage devices.

**SYSLinux** A boot loader for the Linux operating system that promises to simplify the Linux installation.

**Trivial File Transfer Protocol (TFTP)** A set of conventions for transferring files between hosts using minimal protocol.

**VLAN** Short for virtual LAN. A network of nodes that behave as if they were connected to the same cable even though they may actually be physically located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible.

# Index

## A

- adapters
  - ASMA 91, 212
  - Cisco 84
  - ethernet 27
  - Gigabit Ethernet 32, 54
  - MPA 67, 84, 91, 201
  - Myrinet 12
  - RSA 11, 14, 32, 34, 91, 212
  - ServeRAID 97
- addclusteruser 120
- Advanced System Management Adapter
  - see ASMA
- alert timestamp 48
- APC 96
  - apc.tab 202
- apc.tab 202
- apcp.tab 203
- ASMA
  - setup 212

## B

- Beowulf
  - logical view 3
  - parallelism 2
- BIOS
  - COM A 32
  - COM B 32
  - MAC address 83, 100
  - update 97
  - utility 99

## C

- C2T 13, 35
- Cable Chain Technology
  - see C2T
- cables
  - C2T 13, 35
  - Cisco 84
  - compute node 41
  - KVM 13
  - management node 40

- Myrinet 36, 224
- RS485 35
- terminal server 36, 94, 213
- Cisco
  - cisco3500.tab 205
  - MAC address collection 68, 103
  - management VLAN 25
  - setup 84
- cisco3500.tab 205
- cluster
  - Beowulf 3
  - definition 2
  - IBM eServer Cluster 1300 5
  - IBM eServer Cluster 1350 6
  - logical functions 7
  - making up a cluster 7
  - xSeries custom cluster 4
- cluster components
  - ASMA 212
  - Cisco 25
  - ethernet 12
  - Gigabit Ethernet 32, 36, 54
  - KVM switch 13
  - Myrinet 12
  - RSA 14
  - terminal server 13
- Cluster System Management
  - see CSM
- cluster VLAN
  - configuration 48
  - ethernet switch 12
  - IP addresses assignment 23
  - network setup 85
- compute node
  - cables 41
  - definition 8, 11
  - hardware installation 32
  - installation 28
- conserver 46, 69
  - configuration 77
  - conserver.cf 69, 216, 218
  - conserver.tab 70, 208
  - download 77
- conserver.tab 208

control node 9  
CSM 15

## D

### DHCP

APC 96  
configuration 78  
dhcpd.conf 140, 191, 213  
ESP 215  
MAC address collection 27, 100  
makedhcp 78  
MPA 92  
Myrinet 221  
node installation 28  
site definition 57  
version 191

### DNS

configuration 77

### drivers

e1000 54  
ESP 212  
espx 213  
GM 224  
storage node 12  
third party 50, 54

## E

e1000 driver 54  
ELS 20, 69, 93  
Equinox  
    see terminal server  
ESP 96, 210  
    espcfg 214  
    espx 214  
    remote console support 20  
espcfg 214

## F

firewall 48

## G

General Parallel File System  
    see GPFS  
Gigabit Ethernet 32, 36  
    setup 54  
GM 190  
    gmmaker 222

GMVER 223  
installation 222  
makegmroutes 224

GPFS 6

GRUB 47

## H

### hardware setup

cabling 33, 40  
Gigabit Ethernet 36  
KVM 34  
management node 40  
Myrinet 36  
PDU 33  
rack 33  
RSA 32  
terminal server 36  
UPS 33

### High-Performance Computing

    see HPC

HPC 1

## I

### IBM eServer Cluster 1300

CSM 5  
GPFS 6  
overview 5

### IBM eServer Cluster 1350

overview 6

### IBM Linux clusters

IBM eServer Cluster 1300 5  
IBM eServer Cluster 1350 6  
xSeries custom cluster 4

install node 9

### Inter-Process Communication

    see IPC VLAN

### IPC

Myrinet 25

IPC VLAN 23

### iTouch

    see terminal server  
    setupitouch 218

## K

### kernel

custom 109  
tarball 109

- update 51, 240
- Keyboard, video and mouse
  - see KVM

- KVM
  - cables 13

## L

- Linpack 232

## M

- MAC address

- Cisco 68, 103
  - collection 83, 100
  - DHCP 34
  - mac.tab 204
  - nodehm.tab 199
  - populate tables 59
  - site.tab 190
  - terminal server 27, 213

- mac.tab 204

- makedhcp 78, 92, 96, 213

- management node 8

- cables 40

- DHCP 26

- DNS 26

- installation 26

- NFS 26, 74

- NTP 26

- SNMP 27

- SSH 27

- TFTP 26, 28

- management node services

- conserver 57

- Management Processor Network

- see MPN

- management VLAN 23, 25

- Cisco 25

- configuration 48

- Myrinet 25

- Model 330 11, 32, 40

- Model 342 11, 32, 40, 44

- MPA 84

- configuration 67

- mpa.tab 201

- mpasetup 130

- setup 91

- mpa.tab 201

- mpacheck 123

- mpareset 126

- mpascan 128

- mpasetup 130

- MPI/MPICH

- mpimaker 227

- MPN

- definition 14

- RS485 14

- RSA 14, 66

- topology 66

- Myrinet

- adapter 32

- cable 36

- cables 36, 224

- GM 223

- GM driver 222

- gmmaker 222

- GMVER 223

- IPC 25

- makegmroutes 224

- management VLAN 25

- overview 12, 218

- setup 221

## N

- network

- cluster VLAN 48

- configuration 47

- management VLAN 48, 78

- public VLAN 24, 47, 78, 85

- NFS 28

- configuration 74

- node installation 108

- NIS 120, 191, 226

- nodehm.tab 197

- nodelist.tab 193

- nodels 133

- noderange 134

- nodeset 140

- nodetype.tab 196

- NTP

- configuration 75

## P

- passwd.tab 206

- PBS 21

- planning

- cluster VLAN 23
- IPC VLAN 23
- management VLAN 23, 25
- naming convention 22
- public VLAN 24
- xCAT 22
- populate tables 59
- post-installation steps 50
- pping 145
- prcp 147
- prsync 149
- psh 151
- public VLAN 85
  - configuration 47
- PXE 19
  - node installation 28

## R

- rcons 153
- Red Hat
  - bootloader 47
  - Disk Druid 46
  - errata 51
  - firewall 48
  - GRUB 47
  - installation 45
  - network configuration 47
  - package groups 49
  - partitioning 46
  - post-installation steps 50
  - time zone 48
- Redbooks Web site 240
  - Contact us xxv
- Remote Supervisor Adapter
  - see RSA
- reventlog 155
- rinstall 158
- rinv 160
- rpower 163
- rreset 166
- RS485 35
  - mpascan 128
  - MPN 14
- RSA 212
  - AC adapter 34
  - cable 35
  - installation on x330 32
  - setup 32, 85, 91

- rtel.tab 209
- rvid 168
- rvitals 171

## S

- ServeRAID 97
- site.tab 188
- SNMP
  - alert timestamp 48
  - alerts 14
  - configuration 73
  - MPA 91
- SSH 76
  - configuration 76
  - gensshkeys 76
  - keys 114
- stage 1 - hardware setup 84
- stage 2 - MAC address collection 100
- stage 3 - management processor setup 103
- stage 4 - node installation 107
- storage node 9
  - drivers 12

## T

- tables
  - apc.tab 203
- terminal server 13
  - cables 94, 213
  - ELS 20, 69, 94
  - ESP 20, 212
  - iTouch
  - MAC address collecting 27
  - rtel.tab 209
  - setup 93, 212
  - tty.tab 210
- terminal server ESP 96
- TFTP 28, 74
  - configuration 74
- time zone 48
- tty.tab 210

## U

- user node 9

## W

- wcons 174
- winstall 110, 177



wkill 180  
wvid 182

## X

x330 11, 32, 40

x342 11, 32, 40, 44

xCAT 19

    compute node installation 28

    directory structure 20

    download 20

    installation 58

    kstmp 107

    management node installation 26

    planning 22

    populate tables 58

    stage 1 84

    stage 2 100

    stage 3 103

    stage 4 107

    version xxi

    winstall 110

xCAT commands

    addclusteruser 120

    mpacheck 123

    mpareset 126

    mpascan 128

    mpasetup 130

    nodels 133

    noderange 134

    nodeset 140

    pping 145

    prcp 147

    prsync 149

    psh 151

    rcons 153

    reventlog 155

    rinstall 158

    rinv 160

    rpower 163

    rreset 166

    rvid 168

    rvitals 171

    wcons 174

    winstall 177

    wkill 180

    wvid 182

xCAT tables

    apc.tab 202

    apcp.tab 203

    cisco3500.tab 205

    conserver.tab 208

    mac.tab 204

    mpa.tab 201

    nodehm.tab 197

    odelist.tab 193

    noderes.tab 194

    nodetype.tab 196

    passwd.tab 206

    rtel.tab 209

    site.tab 188

    tty.tab 210





**Redbooks**

## Building a Linux HPC Cluster with xCAT







# Building a Linux HPC Cluster with xCAT



**Redbooks**

## **Cluster installation with xCAT 1.1.0 Extreme Cluster Administration Toolkit**

This redbook will guide system architects and systems engineers toward a basic understanding of cluster technology, terminology, and the installation of a Linux High-Performance Computing (HPC) cluster (a Beowulf type of cluster) into an IBM @server Cluster 1300/Cluster 1350.

## **Linux clustering based on IBM eServer xSeries**

This redbook focus on xCAT Version 1.1.0 (Extreme Cluster Administration Toolkit) for installation and administration. All nodes and components of the cluster, such as compute nodes and management nodes, are installed with xCAT. This toolkit is a collection of scripts, tables, and commands used to build and administer a Beowulf type of cluster or a farm of replicated nodes.

## **Red Hat Linux 7.3**

xCAT commands and configuration files are explained in the appendixes of the redbook. Detailed procedures on how to properly configure the Red Hat Linux 7.3 operating system in the nodes of an HPC cluster are also presented.

## **INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION**

### **BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE**

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

**For more information:**  
[ibm.com/redbooks](http://ibm.com/redbooks)