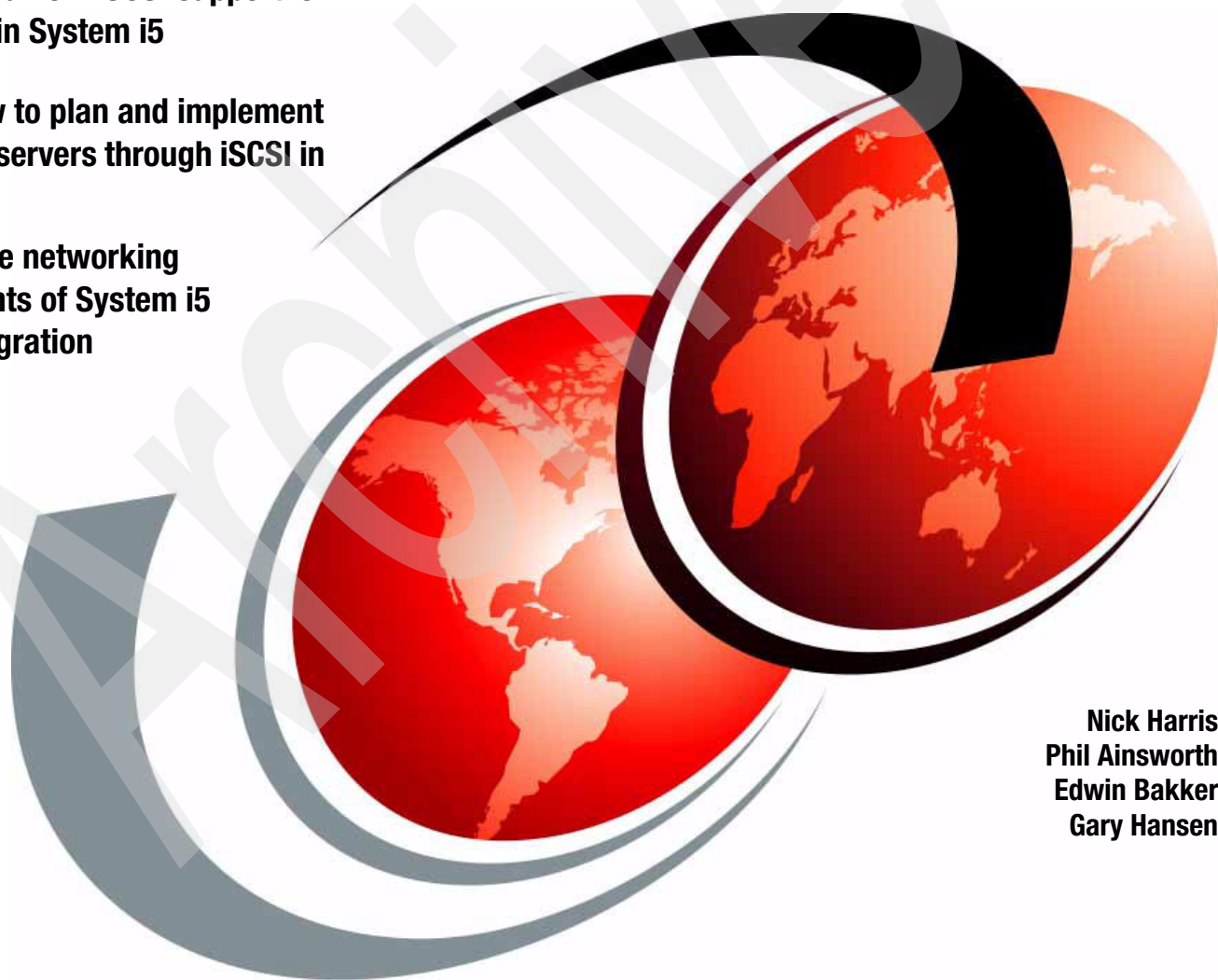


Implementing Integrated Windows Server through iSCSI to System i5

Understand how iSCSI support is available in System i5

Learn how to plan and implement Windows servers through iSCSI in System i5

Review the networking components of System i5 iSCSI integration



Nick Harris
Phil Ainsworth
Edwin Bakker
Gary Hansen

Redbooks



International Technical Support Organization

**Implementing Integrated Windows Server through
iSCSI to System i5**

October 2006

Archived

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

First Edition (October 2006)

This edition applies to Version 5, Release 4, Modification 0 of i5/OS (product number 5722-SS1).

© Copyright International Business Machines Corporation 2006. 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

Notices	ix
Trademarks	x
Preface	xi
The team that wrote this redbook.	xi
Become a published author	xii
Comments welcome.	xii
Chapter 1. Introduction to iSCSI integrated server support	1
1.1 Overview of the content	2
1.1.1 New for V5R4M0.	2
1.2 What is common to iSCSI, IXS, and IXA.	3
1.3 What is iSCSI	5
1.3.1 iSCSI terminology and concepts.	6
Chapter 2. iSCSI architecture on System i5	9
2.1 Introduction to iSCSI on System i5	10
2.2 Component overview	11
2.2.1 Hardware components	11
2.2.2 iSCSI hardware terminology	12
2.3 iSCSI network basics	13
2.3.1 Sending SCSI and VE LAN data over an iSCSI network	14
2.3.2 Comparing targets, initiators, and ports	16
2.3.3 The basic iSCSI configuration.	17
2.4 Objects that define the iSCSI network	19
2.4.1 Objects for a single hosted server.	19
2.4.2 Objects for multiple hosted servers.	22
2.5 iSCSI object descriptions	23
2.5.1 Network server description (NWSD)	23
2.5.2 Network server host adapter.	24
2.5.3 Remote system configuration	27
2.5.4 Service processor configuration	28
2.5.5 The role of IBM Director	29
2.6 Other components of the iSCSI network.	29
2.6.1 Network server storage spaces.	29
2.6.2 IBM Director	30
2.7 iSCSI security model.	30
2.7.1 Service processor connection security	31
2.7.2 iSCSI network security	31
2.7.3 Service processor password.	31
2.7.4 Challenge Handshake Authentication Protocol (CHAP)	32
2.7.5 Network isolation and physical security	32
2.8 IP addressing structure	32
2.9 Hot spare.	34
2.9.1 Integrated DHCP server	35
2.10 General performance considerations	37
2.10.1 System i5 storage spaces.	37
2.10.2 iSeries storage spaces versus dedicated disks	37
2.10.3 iSCSI server performance.	38

2.10.4	Managing hosted system iSCSI adapter utilization.	39
2.10.5	High bandwidth and low latency is desirable for the iSCSI network.	39
2.10.6	Maximum transmission unit (MTU) considerations.	39
Chapter 3.	Planning for iSCSI attached servers.	43
3.1	Understanding the environment	44
3.1.1	Ordering the hardware	44
3.1.2	Planning for System i5 hardware	45
3.1.3	Planning for i5 Software and Preparation	47
3.1.4	Planning for xSeries	47
3.1.5	Planning for BladeCenter	49
3.2	Resources for planning	50
Chapter 4.	Installing the iSCSI integrated server.	53
4.1	Pre-installation requirements	54
4.2	The install Windows server command	54
4.2.1	Objects created by the install command on System i5	55
4.2.2	New configuration objects required for iSCSI	59
4.2.3	The network server configuration objects	65
4.2.4	Reasons for precreating *nwscfg objects prior to the install	77
4.2.5	Install types	77
4.2.6	Removing System i5 objects.	77
4.2.7	Installation.	78
Chapter 5.	Implementing IBM Director Server	107
5.1	Introduction to IBM Virtualization Engine	108
5.2	IBM Director Server in an iSCSI environment.	108
5.2.1	Why IBM Director Server	109
5.2.2	Start IBM Director Server	109
5.2.3	iSeries Hardware requirements.	112
5.2.4	iSeries software requirements.	113
5.2.5	IBM Director Agent (5722-DA1) installed	113
5.2.6	IBM Director Server already installed or older version	114
5.3	Fastpath install	114
5.3.1	Uninstall IBM Director Server on i5/OS using the Uninstaller Launchpad	129
5.3.2	Uninstall IBM Director Server using DLTLCIPGM on i5/OS	135
5.4	For Install failures	136
5.4.1	Steps to display install log files using iSeries Navigator	136
5.4.2	Display install log files using CL command line	137
5.5	For uninstall failures	138
5.6	IBM Virtualization Engine Base Support	138
5.6.1	IBM Director server update	139
5.7	iSCSI integrated server in multiple partitions	140
5.7.1	Create a new user on RSA II service processor	141
5.7.2	Create a new user on MM service processor	143
5.8	QSHHELL commands	145
Chapter 6.	Managing integrated iSCSI environments	149
6.1	Introduction to management concepts	150
6.2	Choose your two possible interfaces	154
6.3	Working with iSCSI integrated server	155
6.4	Manage network server host adapters	156
6.4.1	Manage NWSH objects using iSeries Navigator.	156
6.4.2	Manage NWSH objects using CL commands.	163

6.5	Manage service processor server configuration	169
6.5.1	Initialize options for user ID and Password on SP/MM	169
6.5.2	Initialize new or changed SP configuration using iSeries Navigator	172
6.5.3	Initialize new user ID and Password on SP/MM using iSeries Navigator	172
6.5.4	Get user ID and password in sync with SP/MM and SRCVPRC object	173
6.5.5	Change service processor password using iSeries Navigator	174
6.5.6	Initialize new or changed SP configuration using CL command	175
6.5.7	Create new user ID and Password on SP/MM using CL command	176
6.5.8	Synchronize user ID and password with SP/MM and SRCVPRC object	176
6.5.9	Change service processor user ID and password using CL command	176
6.5.10	Manage service processor server configurations using iSeries Navigator	177
6.5.11	IP address change for Service Processor.	181
6.5.12	Manage service processor server configurations using CL command	182
6.6	Manage remote system server configurations	185
6.6.1	Manage RMTSYS object using iSeries Navigator.	185
6.6.2	Manage RMTSYS object using CL command.	195
6.7	Manage connection security configurations	202
6.7.1	Manage connection security configuration using iSeries Navigator	202
6.7.2	Manage connection security configuration using CL command	205
6.7.3	Manage network server description (NWSD) using iSeries Navigator	207
6.8	Starting iSCSI integrated server	218
6.8.1	Starting iSCSI integrated server during an iSeries IPL	219
6.8.2	Starting iSCSI integrated server within a CL Program	220
6.8.3	Starting iSCSI integrated server from Service Processor	221
6.8.4	Starting iSCSI integrated server from iSeries Navigator	224
6.8.5	Starting iSCSI integrated server using CL command	228
6.8.6	Starting Remote Control on MM to see Windows booting	229
6.8.7	Starting Remote Control on RSA II to see Windows booting	230
6.9	Stopping iSCSI integrated server	231
6.9.1	Stopping an iSCSI integrated server using Windows Desktop	232
6.9.2	Stop/Restart iSCSI integrated server using iSeries Navigator	233
6.9.3	Stopping an iSCSI integrated server using CL commands.	235
6.9.4	Stopping integrated servers using a CL Program	236
6.9.5	Stopping an iSCSI integrated server using the MM Web Interface.	236
6.9.6	Stopping an iSCSI integrated server using the RSA II Web Interface	237
6.10	Status iSCSI integrated server	237
6.10.1	Status of iSCSI integrated servers using iSeries Navigator	238
6.10.2	Status of iSCSI integrated servers using CL commands	239
6.10.3	xSeries or BladeCenter status using iSeries Navigator/CL commands	240
6.10.4	Status of Blade using Web Interface.	241
6.10.5	Status of xSeries using Web Interface	241
6.10.6	Status of the NWSD degraded	241
6.11	Manage Storage Spaces with iSCSI integrated server.	241
6.11.1	Expanding a system drive (C:) using iSeries Navigator	241
6.11.2	Expanding a system drive (C:) having multiple integrated servers	245
6.11.3	Expand a Disk (non-system) using iSeries Navigator	246
6.11.4	Expanding system drive (C:) using CL command	246
6.11.5	Expanding a system drive (C:) having more integrated servers	250
6.11.6	Expand a Disk (non-system) using CL command.	251
6.11.7	Add (link) disk drive using iSeries Navigator.	251
6.11.8	Copy a storage space using iSeries Navigator	252
6.11.9	Unlink disk drive using iSeries Navigator	252
6.11.10	Delete disk drive using iSeries Navigator	253

6.12 Change Network Host Server Adapter procedure.	254
6.12.1 Change NWSH for server storage spaces using iSeries Navigator	254
6.12.2 Change NWSH for Virtual Ethernet connection	254
Chapter 7. Backup and Recovery	255
7.1 Overview of backup and recovery.	256
7.1.1 Backup from a server-centric perspective.	257
7.2 Planning a backup strategy.	258
7.2.1 Staged backup	258
7.2.2 Backup and recovery tips	259
7.2.3 Automating backup and recovery	261
7.2.4 Backup technique positioning and recommendations.	261
7.2.5 Recommended backup schedule	262
7.2.6 Hot spare	263
7.3 i5/OS-centric backup.	265
7.3.1 Storage space backup overview	266
7.3.2 Storage space backup tips	267
7.3.3 Components of the Windows integration environment	267
7.3.4 Using BRMS to back up an integrated Windows server	269
7.3.5 i5/OS-centric backup methods	270
7.3.6 i5/OS-centric backup using the Save menu	270
7.3.7 i5/OS-centric backup using CL commands.	272
7.3.8 i5/OS-centric backup using CL programs	275
7.4 i5/OS-centric recovery.	278
7.4.1 Integrated hardware considerations	279
7.4.2 i5/OS-centric recovery roadmap	280
7.4.3 i5/OS-centric recovery methods	282
7.4.4 i5/OS-centric recovery using CL commands.	283
7.4.5 i5/OS-centric recovery using CL programs	286
7.4.6 File level recovery via storage spaces	289
7.5 Windows-centric backup and recovery	290
7.5.1 Overview	290
7.5.2 Windows recovery options	292
7.5.3 Choosing a tape drive for use by your Windows backup application	293
7.5.4 Restricting System i5 tape drives that can be used by Windows	293
7.5.5 Setting up a System i5 tape drive for use by Windows.	293
7.5.6 Backing up Windows files to the System i5 IFS	299
Chapter 8. Virtual Ethernet LAN	305
8.1 Introduction to Virtual Ethernet LAN (VE LAN)	306
8.1.1 Basic concepts	306
8.1.2 Virtual LAN benefits	307
8.1.3 Virtual LAN limitations.	307
8.2 Types of Virtual Ethernet LANs.	307
8.2.1 Point-to-point Virtual Ethernet LAN.	307
8.2.2 Non-point-to-point VE LANs	309
8.3 Inter-partition connections.	309
8.3.1 Inter-partition networks with the Hardware Management Console	309
8.4 VE LAN configuration examples	310
8.4.1 Server to non-hosting partition (inter-partition) VE LAN scenario	310
8.4.2 Server to server VE LAN scenario - different hosting partitions	312
8.4.3 Complex inter-partition VE LAN scenario	313
8.5 Virtual Ethernet LAN IP addressing considerations	314

8.6 Virtual Ethernet performance considerations	315
8.7 Setting up VE LAN connections	315
8.7.1 Creating VE LAN resources using the HMC	315
8.7.2 Creating a Virtual Ethernet resource using the HMC	315
8.7.3 Setting up a VE LAN connection in an i5/OS partition	321
8.8 Browsing the Virtual Ethernet LAN topology	322
8.8.1 View Ethernet connections from i5/OS	322
8.8.2 View Ethernet connections from the integrated Windows server console	323
Chapter 9. Scaling your iSCSI network	325
9.1 Overview	327
9.1.1 Terminology	327
9.1.2 Introduction to scaling the iSCSI network	330
9.2 Increasing the bandwidth of a hosted server connection	331
9.2.1 Increasing the bandwidth of a hosted xSeries server connection	331
9.2.2 Increasing the bandwidth of a hosted Blade server connection	335
9.3 Increasing the number of hosted servers	340
9.4 iSCSI configuration nomenclature.	341
9.5 Scenarios for scaling a hosted xSeries server connection	343
9.5.1 xSeries configuration 1: The basic iSCSI configuration	343
9.5.2 xSeries configuration 2: Sharing a target HBA between hosted servers	345
9.5.3 xSeries configuration 3: Adding bandwidth to a hosted server	346
9.5.4 xSeries configuration 4: Adding another basic iSCSI configuration	349
9.5.5 xSeries configuration 5: Splitting the workload between target HBAs	351
9.6 Scenarios for scaling a hosted Blade server connection	352
9.6.1 Blade configuration 1: The basic iSCSI configuration	353
9.6.2 Blade configuration 2: Sharing a target HBA between hosted servers	354
9.6.3 Blade configuration 3: Adding bandwidth to a hosted server	355
9.6.4 Blade configuration 4: Adding another basic iSCSI configuration	358
9.6.5 Blade configuration 5: Splitting the workload across multiple data paths	360
9.7 Introduction to path deployment	363
9.7.1 Overview	363
9.7.2 Storage paths	364
9.7.3 Path deployment considerations	367
9.8 Automatic path deployment	369
9.8.1 Automatic path deployment (physical view) - xSeries server	370
9.8.2 Automatic path deployment (data flow view) - xSeries server	372
9.8.3 Automatic path deployment (physical view) - Blade server	374
9.8.4 Automatic path deployment (data flow view) - Blade server	375
9.9 Manual path deployment - target side	377
9.10 Manual path deployment - initiator side	379
9.10.1 Overview	379
9.10.2 Manually deploying data paths on the initiator side - xSeries server	380
9.10.3 Manual path deployment (data flow view) - xSeries server	382
9.10.4 Manually deploying data paths on the initiator side - Blade server	384
9.10.5 Manual path deployment (data flow view) - Blade server	385
9.11 Capacity planning for iSCSI	387
9.11.1 Introduction	387
9.11.2 Getting started with capacity planning for iSCSI	391
9.11.3 Obtain Windows performance data	392
9.11.4 Calculate the number of target HBAs required	395
9.11.5 Calculate CPW requirements	396
9.11.6 Calculate memory requirements	397

9.11.7 Calculate disk requirements	398
9.11.8 Scalability limits.	399
9.12 IP addressing in a System i iSCSI network.	401
9.12.1 iSCSI network addressing.	401
9.12.2 Choosing an IP addressing schema	404
9.13 Creating an iSCSI connections map	405
9.13.1 Running the QVNIMAP command	405
9.13.2 Using the QVNIMAP switches.	406
9.14 Scaling tasks	417
9.14.1 Installing an additional target HBA in the hosting partition	419
9.14.2 Configuring an additional target HBA in the hosting partition	419
9.14.3 Creating a storage path using the additional target HBA	423
9.14.4 Installing an additional initiator HBA in an xSeries server.	425
9.14.5 Configuring an additional initiator HBA port in an xSeries server	426
9.14.6 Configuring the additional initiator HBA port as a non-boot port	426
9.14.7 Configuring the additional initiator HBA port as the boot port	429
9.14.8 Updating the RMTSYS configuration object	433
9.14.9 Configuring the second initiator HBA port in a Blade server	436
9.15 Path deployment tasks	436
9.15.1 Redeploying a storage space to a different target HBA	437
9.15.2 Redeploying a Virtual Ethernet LAN to a different target HBA	441
9.15.3 Changing the boot port	443
9.15.4 Preventing a non-boot drive from using an initiator HBA port	444
9.15.5 Redeploying a Virtual Ethernet LAN to a different initiator HBA port	446
Related publications	453
IBM Redbooks and Redpapers	453
Online resources	453
How to get IBM Redbooks	454
Help from IBM	454
Index	455

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.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®
Redbooks (logo) ™
eServer™
ibm.com®
iSeries™
i5/OS®
xSeries®
AIX®

AS/400®
BladeCenter®
IBM®
NetServer™
OpenPower™
OS/400®
PowerPC®
Redbooks™

System i™
System i5™
System p5™
System x™
System Storage™
Tivoli®
Virtualization Engine™

The following terms are trademarks of other companies:

Java, JRE, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Excel, Internet Explorer, Microsoft, Windows Server, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbook discusses the newest implementation of the Integrated xSeries® servers, which utilizes the iSCSI protocol to communicate with the System i5™. The iSCSI product offers an additional scalable network connection to the System i5 system bus or the HSL loop. In this IBM Redbook, we discuss:

- ▶ The architecture and scenarios for utilizing this new function
- ▶ Concepts and terminology of iSCSI
- ▶ Planning for iSCSI environment
- ▶ IBM Director Server and iSCSI
- ▶ Operating iSCSI Servers
- ▶ Backup and Recovery of iSCSI attached servers
- ▶ Virtual Ethernet in an iSCSI environment

This IBM Redbook offers planning and implementation advice and guidance for IBM, IBM Business Partners, and customers.

The team that wrote this redbook

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

Nick Harris is a Consulting IT Specialist for the iSeries™ and has spent the last eight years in the International Technical Support Organization, Rochester Center. He specializes in eServer™ i5 and iSeries hardware, i5/OS® and OS/400® software, LPAR, High Availability, external disk, Windows® integration, and Linux®. He writes IBM Redbooks™ and teaches IBM classes at ITSO Technical forums worldwide on all these areas and their relationship to system design and server consolidation. He spent 13 years in the United Kingdom (UK) AS/400® Business, where he worked with S/36, S/38, AS/400, and iSeries servers. He can be contacted at <mailto:nihharris@us.ibm.com>

Phil Ainsworth is a Senior Technical Specialist with IBM Australia and has supported the iSeries, AS/400, and their predecessors since 1980. He provides pre-sales technical support for the iSeries marketing team and specializes in Windows integration on iSeries. Other areas of interest include the implementation of Storage Area Networks and Logical Partitioning on iSeries. Since 1988, Phil has co-authored a number of redbooks on AS/400 and iSeries technologies, including all seven redbooks on running Windows and Linux on the integrated xSeries server (IXS), integrated xSeries adapter (IXA), and now Windows integration over iSCSI. He is also a regular presenter at IBM conferences and other technical events. He can be contacted at <mailto:philains@au.ibm.com>

Edwin Bakker has been working with IBM for eight years as an advisory IT Specialist in the Netherlands at iSeries Software Support, supporting software on the iSeries in total but specialized in iSeries Access, Communications, IXS/IXA/iSCSI, and Service Agent. He can be contacted at mailto:edwin_bakker@nl.ibm.com

Gary Hansen is an Advisory Software Engineer. He has over 18 years experience with software support for AS/400, iSeries, and System i5. He has worked with the Integrated Server product since its inception on the FSIOF. He is the team leader for the Integrated

Server Support on the iSeries. He has written and taught classes about various aspects of the Integrated servers. He has worked on two previous IBM Redbooks covering Linux topics. His areas of expertise are client connectivity, iSeries Integrated Server Support, networking, and Linux. He can be contacted at: <mailto:hansengl@us.ibm.com>

The team would like to extend a special thanks to Mike Schambureck, of the iTC, Rochester Development Lab. Mike supplied all the equipment, co-delivered the three day workshop we attended, solved our hardware requests, and has always been ready to answer questions and give advice.

We would like to thank the following from the IBM Rochester Development Lab:

Josep Cors, George Gaylord, Jeff Meaden, Randy Nelson, and Kyle Wurgler

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

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

- Send your comments in an email to:

redbook@us.ibm.com

- Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400



Introduction to iSCSI integrated server support

The iSCSI Integrated xSeries server is the newest implementation of the integrated xSeries server family. This chapter will look at:

- ▶ What is covered in the book.
- ▶ What is new with Integrated Server Support for System i5.
- ▶ iSCSI and iSCSI terminology.
- ▶ Components of the iSCSI implementation for System i5, xSeries, and the Blade Center.
- ▶ How iSCSI can benefit your environment.

1.1 Overview of the content

Over a number of years, the System i5 has evolved from an integrated operating system that incorporated functions into an operating system with built-in database and security functions, which simplifies the installation and management of various OS pieces into a system. The System i5 can function as a Storage Area Network for various platforms, including Microsoft® servers and clients, Linux servers and clients, and AIX® servers. It offers functions such as the:

- ▶ Integrated File System and QNTC and NFS file systems in particular
- ▶ iSeries Netserver
- ▶ Integrated Server support for Intel® servers running Windows for Linux and PowerPC® Servers in Linux and AIX partitions.

This document discusses the newest implementation of the Integrated xSeries servers, which utilizes the iSCSI protocol to communicate with the System i5. The iSCSI product offers an additional scalable network connection to the System i5 system bus or the HSL loop. In this Redbook, we will discuss:

- ▶ The architecture and scenarios for utilizing this new function
- ▶ Concepts and terminology of iSCSI
- ▶ Planning for the iSCSI environment
- ▶ IBM Director Server and iSCSI
- ▶ Operating iSCSI Servers
- ▶ Backup and Recovery of iSCSI attached servers
- ▶ Virtual Ethernet in an iSCSI environment

1.1.1 New for V5R4M0

Much of what is new about the iSCSI Server is related to the hardware implementation and the configuration required to implement that hardware. Certainly there are additional capabilities that the iSCSI implementation provides, but these are mainly related to utilizing disk, Virtual Ethernet, and how the system is started and stopped. The actual integration of applications is not essentially different from the user's perspective than those which are available to the current IXS and IXA servers and those at previous releases.

There are several new features with V5R4M0 for Integrated Server Support and new enhancements to iSeries Navigator. We mention them here, because some of the new features enhance the iSCSI environment and you need to be aware of them for implementing iSCSI support. Changes for V5R4M0 include:

1. The licensed program product (LPP) was repackaged from 5722-WSV (see Figure 1-1) to 5722-SS1 option 29 (see Figure 1-2 on page 3). There is a new LPP for Linux, but the base support for Intel Linux is part of the Integrated Server Support.

Note: Linux is not supported on the iSCSI integrated server at this time.

5722WSV	*BASE	IBM Integration for Windows Server
5722WSV	2	Integration for Windows 2000 and 2003

Figure 1-1 Licensed Programs for the integrated server V5R3M0

5722SS1	29	Integrated Server Support
5722LSV	*BASE	IBM i5/OS Integration for Linux on xSeries

Figure 1-2 Licensed Programs for Integrated Server in V5R4M0

2. iSCSI Integrated xSeries server support is introduced.
3. Support for integrating IBM Blade Centers with iSeries using iSCSI connections.
4. iSCSI integrated servers require that you install IBM Director Server. IBM Director Server itself has other software requirements as well. See Chapter 5, "Implementing IBM Director Server" on page 107.
5. iSCSI servers support larger storage capacity - up to 63 storage spaces that can be 1 Terabyte in size.
6. The ability to expand storage spaces without copying them as in previous releases. This is probably easier to do with iSeries Navigator, but you can use the new CL command CHGNWSSTG alternatively. For more discussion, see:

<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/index.jsp?topic=/rzahq/rzahq/integservov.htm>

7. Additional integration services in Windows to manage shutdown and Ethernet connections.
8. iSeries Navigator has changed so that the Integrated Server Administration view is now in the root view and has the capability of administering all of the servers that integrate with the System i5, including hosted AIX and Linux Servers. In addition, you can configure Virtual Ethernet for the integrated servers. This support is useful for creating some of the devices associated with the iSCSI integrated server.
9. Support for Windows Server® 2003 Volume Shadow Copy Service was added to the file level backup through Qntc. Qntc itself was modernized to use the CIFS protocol by default rather than SMB which should make this file system easier to manage.
10. Support was removed for all of the IXS server hardware that is older than the 2890 IXS.

1.2 What is common to iSCSI, IXS, and IXA

In the previous section, we stated that the integration application for iSCSI is not essentially different than that which is provided for IXS and IXA. By this we mean that the administration and management capability is the same on all of the three hardware platforms:

1. All servers are installed using the INSWNTSVR command.
2. They all must be diskless and thus use virtual disk served from System i5, but also they can use virtual optical and tape (share System i5 tape and optical drives).
3. They are all started and stopped from iSeries.
4. They run the Windows operating system within which are installed drivers needed to share iSeries hardware, and services that provide management capabilities such as:
 - a. Enrollment of System i5 users and groups into the Windows operating system
 - b. Synchronization of passwords for users who have been enrolled
 - c. The ability to submit commands from iSeries to the Windows operating system (SBMNWSCMD)
 - d. Various options for saving the operating system and user data

- i. Shared Tape running the Windows backup utility or supported third party backup applications
- ii. File level backup through QNTC enhanced as per the note above, see 1.1.1, "New for V5R4M0" on page 2
- iii. System i5 SAV/RST for disaster recovery
- e. Log Windows event log messages on the iSeries.
- f. Report some Windows statistics to the iSeries.

IXS, IXA, and iSCSI

If the three hardware platforms offer essentially the same features, what is the requirement to have iSCSI connection?

The IXS, which was the original integrated server, was installed into the System i5 towers. It was often referred to as blade-like, because it was housed inside the iSeries in a manner similar to that of a Blade. It lacked a diskette drive, hard drive, and optical drive. It mostly used virtual devices for its hardware except for keyboard, mouse, and display. It eventually had USB ports enabled so it could use a diskette after the Windows OS booted. The hardware was totally controlled by the iSeries.

The IXS hardware is limited to single processor capability. It is sufficient for many common server requirements but lacked the capacity for some installations. The IXA provided the capability to install more powerful servers and to more easily integrate newer models of xSeries servers. Unlike the IXS, it was not housed in the System i5 but was separate.

Both the IXS and IXA had architectural restrictions that limited the number of devices that could be supported:

- ▶ The IXS cards, which were installed into the iSeries, occupied multiple slots on the bus and thus reduced the available slots for other resources. The number of servers that could be installed was limited by the slots available and the number of towers.
- ▶ The IXA was limited by the HSL architecture. There was a limited number of expansion towers in a loop and a limited number of loops per model.
- ▶ Thus, the maximum number of IXS or IXA servers that could be installed in the high end systems was 60. The maximum number of HBA adapters that can be installed with the high end system is 168, and these adapters support up to eight initiator connections.

The chief difference between iSCSI and the other two implementations is that the devices are connected to the iSeries by Ethernet cabling and switches rather than inserted into the bus or connected to the bus via the HSL loop. One target adapter takes up one slot and can support up to eight connections. This provides more scalability and flexibility but also means that the iSeries has to be told what these devices are. Thus, the most notable differences are:

- ▶ The devices are network-connected rather than inserted into the bus structure of the System i5.
- ▶ There is additional configuration required on the System i5 to describe them to the system.
- ▶ The devices are managed for power control using IBM Director Server.
- ▶ There are additional tasks required to configure and manage the devices.

Maybe the most attractive addition that comes from the iSCSI implementation is the ability to provide BladeCenter® support. Although Blade Centers can have DASD, they are designed for use with a SAN. Their packaging is the embodiment of server consolidation; therefore, they are a perfect partner for integration with the iSeries.

A capability that did not exist with IXS or IXA is the ability to shut down the server while it is varied on. The server status will go from active to varied on, but should return to active when the server is brought up again. Unlike the IXA, which had to be powered down and in some cases disconnected from the HSL loop in order to perform maintenance, this is not the case with iSCSI servers.

A noteworthy difference between iSCSI and IXA solutions is that the service processor needs to be accessed from the network in order for the device to be located and varied on. Thus whereas the xSeries server needs to have the default user id and password retained and needed a proprietary RS485 connection, the current implementation uses an Ethernet connection to the service processor and it needs to be accessible via the iSeries network connection.

1.3 What is iSCSI

iSCSI stands for Internet Small Computer System Interfaces. Many readers will recognize the SCSI portion of the acronym as a reference to one of the standard protocols for accessing storage devices and the Internet portion is obvious. iSCSI, then, is a protocol that was and is being developed to use existing TCP/IP infrastructures to transfer block data to networked storage devices (SAN devices). SCSI commands are encapsulated in TCP/IP packets and presented to the storage device in a manner that the devices require. It is described in RFC3720 and RFC3721. For more information see:

<http://www.faqs.org/rfcs/rfc3720.html>

For a list of related rfcs, see:

<http://www.real-storage.com/iscsi-rfc.html>

The iSCSI implementation provides a more cost-effective solution than Fibre Channel for many businesses, because they can use most of the same technology that is already being used for their other business processes. It also allows them to provide SAN access across larger areas and to manage them in a similar manner to the way that they are already managing their network. An additional benefit is that security options are built into the iSCSI specification, whereas Fibre Channel has not been capable of passing security messages. CHAP is available for iSCSI, and IPsec is currently being developed.

Fibre Channel has been desirable for its performance characteristics, but advocates of iSCSI say that on 1 gigabit networks, iSCSI can compete. iSCSI communications take place between a target (storage provider) and an initiator (client).

The integrated server has often been referred to as a SAN implementation because the server hardware does not have its own local hard drives but utilizes the System i5 DASD. This newest implementation uses a special network card referred to as a host bus adapter (HBA) on the iSeries to communicate through a 1 GB switch over a TCP/IP network with the client HBA, which can be installed in either a supported model of xSeries server or an IBM Blade server. The immediate benefit of the new connectivity is that Blade Centers can now be integrated with the iSeries as well as xSeries servers. The fact that the server is Ethernet connected via host bus adapters (HBAs) rather than being a component in the iSeries HSL loop can also be viewed as advantageous because it removes it from any potential conflict with the System i5 expansion towers.

1.3.1 iSCSI terminology and concepts

Depending on which documents you reference, there are a number of terms which can be confusing to the user. We discuss a few of these terms that are relevant to the discussion in this book.

The following terms might confuse those who are not familiar with iSCSI protocol or reference new terms for the iSCSI integrated server.

- ▶ Challenge Handshake Authentication Protocol (CHAP) is an authentication protocol. It uses a secret that is shared between target and initiator. It is configured in the *rmtsys (remote system) configuration object. The target authenticates the initiator request via this means. It is optional, but we highly recommend you set CHAP up.
- ▶ Enclosure ID identifies the enclosure, which contains the service processor. It consists of the serial number, type, and model. For an IBM Blade Center, it is derived from the chassis and not the individual blade.
- ▶ Host bus adapters (HBAs) are adapters that plug into the bus of a system. In this case, they are an adapter for iSeries (target) and for the xSeries or Blade server (initiator) that provide the connectivity for the iSCSI storage requests and Virtual Ethernet traffic. This connectivity is via an Ethernet connection to a 1 GB switch.
- ▶ Hot spare is an extra server that can be interchanged for a failed server. In a Blade center, it might be another Blade that is not being used. Because of the i5 objects, a server of the appropriate hardware type or an HBA can be swapped in by simply changing the hardware resource in the appropriate object.
- ▶ iSCSI qualified name (IQN) is a unique name that identifies a target or initiator as defined in RFC3722. We advise you to choose to generate it.
- ▶ Local interface refers to the iSeries configuration and, thus, to the target adapter. All references to local refer to iSeries.
- ▶ Network server configuration (nwscfg) object. There are three subtypes of this object that represent the initiating server. They are unique to iSCSI and new for this release:
 - Remote system object is of type *rmtsys and defines the connection to the remote system and specifies CHAP and boot information. One per each xSeries which would have multiple interfaces defined if it had more than one HBA. In a Blade Center, there would be one for each blade. It also links to the service processor description.
 - Service processor is of the type *srvprc and identifies the service processor location. It points to the management module for the Blade Center.
 - Connection security network server configuration is of the type *cnec. It has to be configured but is not used at this time so the same *cnsec can be used for all servers. When it is used, it will define specific security characteristics for a server or servers.
- ▶ Network Server Description (NWSD) describes an instance of an integrated server. It contains links to all of the various objects that describe the server. It also is used to power on and off the server.
- ▶ Network Server Host Adapter (NWSH) is a device description that defines a particular target HBA. It is used to vary on the HBA and provides the path for storage and Virtual Ethernet.
- ▶ Network Server Storage Spaces (nwsstg) are files in the System i5 Integrated File System that are presented to the Windows server as its local disks. The server boots from these NWSSTG spaces that appear as the system drive and other virtual disks that store user data.

- ▶ Point-to-point Virtual Ethernet connection is created by the install command and is a private network consisting of two nodes: System i5 and a particular Windows server. It is used to provide communications between the iSeries and the server.
- ▶ Remote interface refers to the initiator configuration. All references to remote refer to the client side.
- ▶ Storage path defines which NWSH services a storage space. One nwsd can be connected to four NWSH devices. Storage paths and Virtual Ethernet paths provide a large degree of flexibility in customizing communications channels to provide the appropriate bandwidth to an application.
- ▶ Target HBA is an adapter that services SCSI requests and Virtual Ethernet requests. It also is referred to as the *host* or the *local adapter*. For the integrated server, it is the iSeries adapter. It reports in to the iSeries as a 573B or 573C depending on whether it is copper. One iSeries target adapter can connect eight initiator ports, although it has one physical port.
- ▶ Initiator HBA is an adapter that is installed in an xSeries server or an IO expansion card that is installed into a Blade server that initiates requests or submits scsi commands to the target adapter. It is also referred to as the *hosted adapter*, the *remote adapter*, or the *integrated server adapter*. An xSeries server can have as many as four HBA cards providing four ports to connect to a target or targets; each Blade can have up to two ports available if the proper hardware configuration exists.
- ▶ TOE is TCP/IP offload engine. The HBA adapter services both SCSI requests and Virtual Ethernet requests. Thus, there are two ip addresses and mac addresses for each adapter. The addresses are on the label of the HBA.
- ▶ MAC - media access control. The layer which controls how data is put on the network. The significance for this publication is that the MAC for iSCSI and TOE have to be configured in the remote system object. The label on the HBA contains two addresses: iSCSI and TOE. The addresses consist of 12 hexadecimal characters.
- ▶ iSCSI network is the network that provides the connection between the target HBAs and the initiator HBAs. This network serves storage (SCSI requests) to the initiator.
- ▶ Service processor is a separate processor that is used to control power to the device and perform some management and diagnostic functions. There are three types of processors available with xSeries servers and blades:
 - Base Management Controller (BMC) is used to control some xSeries servers. It is not able to be used for the iSCSI connection in most cases.
 - Management module is the management processor for the BladeCenter as a whole. Thus, from the perspective of the iSeries configuration, all 14 blades in the chassis have the same Service processor.
 - Remote Supervisor Adapter (RSA) is the processor that is required for many of the xSeries servers. There are three models of this:
 - RSA
 - RSA II
 - RSA II - EXA
- ▶ IBM BladeCenter is a chassis that houses “Blade servers”. The chassis has a reduced form factor that provides an enclosure for up to fourteen blades, each of which is an individual server card. In a 42U rack, six chassis can be fit which would house 84 servers in the rack. The blades are slotted into the chassis, which furnishes some of the hardware, such as blowers, power supplies, network access modules, and so on, that is shared by all of the blades. There are various models of chassis and blades, not all of which are

supported with the iSCSI implementation. For more details about what hardware to purchase or which hardware is supported, see:

<http://www.ibm.com/servers/eserver/series/integratedservers/iscsivermodels/>

- IBM Director Server is an application that is part of the Virtualization Engine™ and which is used by iSCSI to discover servers and power them up, or shut them down.

iSCSI architecture on System i5

This chapter discusses the iSCSI architecture as implemented on the System i5.

The iSCSI architecture on System i5 is different than the previous integration architecture based around the Integrated xSeries Server (IXS) and Integrated xSeries Adapter (IXA), although there are many common functions and components. IXSs and IXAs connected to the iSeries or System i5 over the internal Bus in the case of the IXS, or the external HSL bus in the case of the IXA. In the case of iSCSI, xSeries and Blade servers connect to the System i5 over an iSCSI network. Therefore, the big difference between the iSCSI architecture and IXS/IXA architecture is the mode of communication between the hosting i5/OS partition and the hosted server. Because of this difference, we need new i5/OS objects to describe the iSCSI network.

2.1 Introduction to iSCSI on System i5

At its core, the iSeries and System i5 integration architecture, whether it is using the IXS and IXA or the new iSCSI architecture, provides a SAN capability to connected Intel-based servers. This is the same capability provided by traditional SANs, but with some architectural, functional, and management benefits that are not available with traditional SANs.

The new iSCSI-based SAN architecture as implemented on System i5 is represented in Figure 2-1. The two most obvious architectural features of the System i5 iSCSI SAN implementation are the use of single level storage and industry standard iSCSI networking.

Single level storage is the revolutionary and unique storage management architecture that has always been available on the System i5 and its predecessors. This capability makes available the same ease of management that AS/400, iSeries, and System i5 users have always taken for granted and makes it available to the SAN world.

Using the iSCSI protocol for the transport of SCSI data is an established industry standard, in contrast to HSL which was used with the previous IXS/IXA architecture. iSCSI makes available to System i5 users technology that is commonly available in the marketplace, and standards that will be enhanced over time.

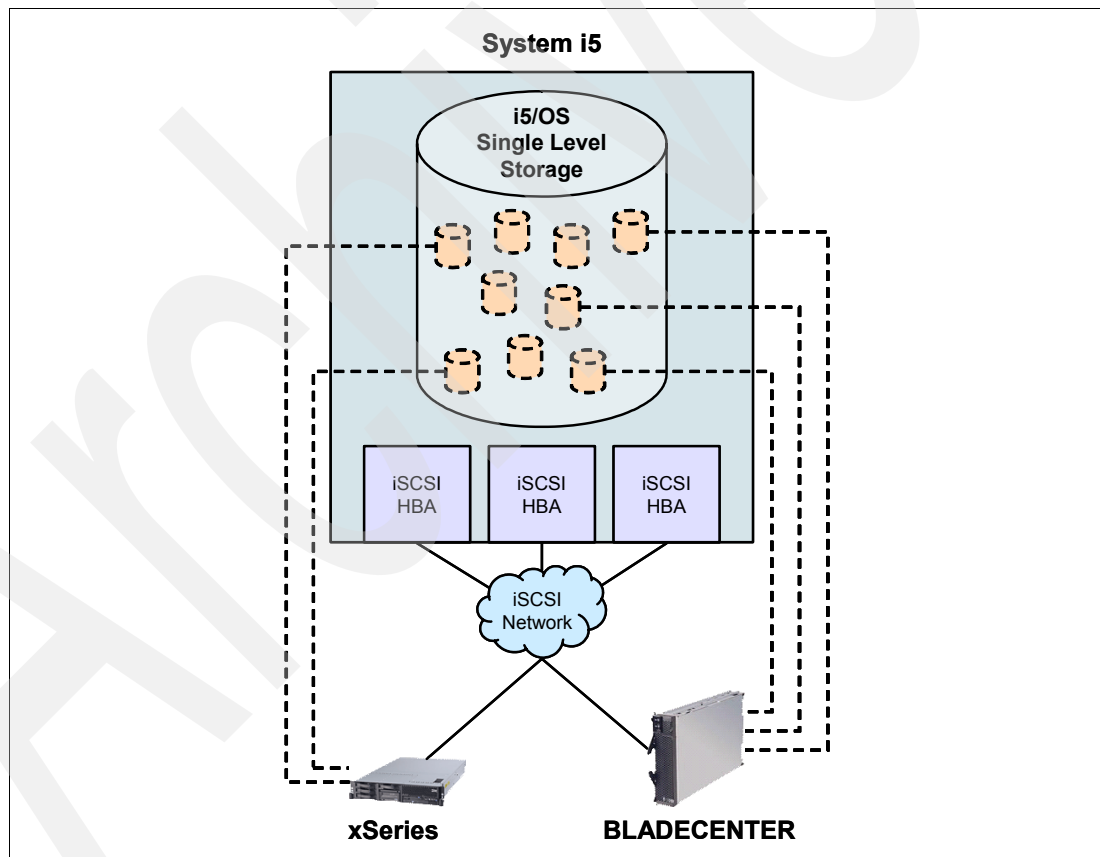


Figure 2-1 System i5 iSCSI SAN architecture

The many benefits to the previous IXS/IXA integration architecture are still available with the new iSCSI implementation; however, the iSCSI architecture provides several important new advantages including:

- ▶ The capability of connecting IBM BladeCenter servers to the System i5 SAN
- ▶ Less rigid limits on the number of servers that can be connected to the System i5 SAN compared with IXS and IXA

Next we discuss the benefits capabilities that were available previously with IXS and IXA and are still available with iSCSI.

2.2 Component overview

Here we compare the hardware requirements on the previous HSL-based IXS/IXA integration architecture with the new iSCSI architecture.

2.2.1 Hardware components

The previous HSL-based IXS/IXA SAN architecture, shown in Figure 2-3 on page 12, enabled xSeries servers to be connected to iSeries and System i5 in two ways:

- ▶ Over the internal bus of the iSeries in the case of IXS
- ▶ Over a 1 GB HSL connection in the case of IXA

Although very fast, this architecture imposes strict limits on the number of IXS and IXA servers that can be connected to the iSeries. IXS servers require special PCI slots in the iSeries system unit and I/O towers. Therefore, to increase the number of IXSs, you might need to acquire an additional I/O tower, which is expensive. Also, HSL connectivity imposes strict limits on the number of xSeries servers that can be connected externally to iSeries. Also, BladeCenter connectivity over HSL is not supported, which also restricts the flexibility of this implementation.

Despite these restrictions, many customers have installed the IXS/IXA integration technology for the many benefits it provides.

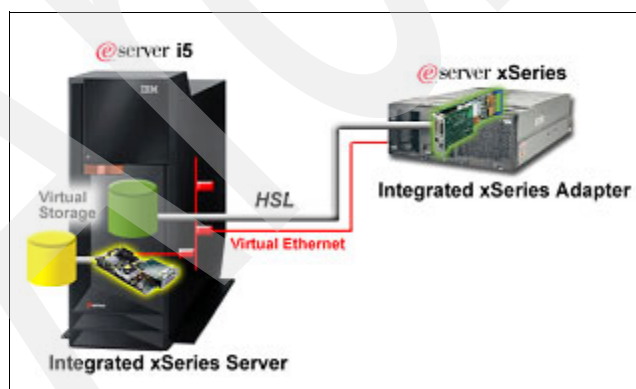


Figure 2-2 HSL-connected xSeries servers

The new industry standard iSCSI-based SAN implementation, as shown in Figure 2-4 on page 13, reduces the previous limitations on the number of servers that can be connected to System i5 and provides full BladeCenter connectivity. Therefore, it is better able to meet the SAN requirements of System i5 customers.

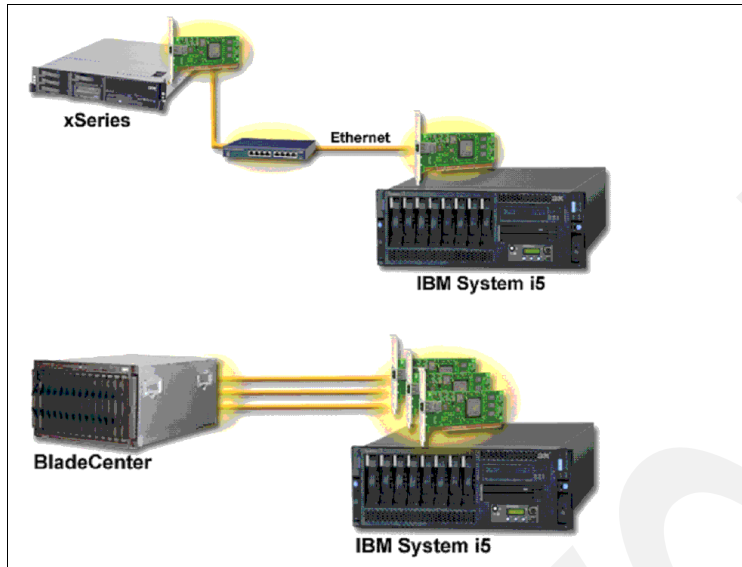


Figure 2-3 iSCSI-connected xSeries and BladeCenter servers

The key hardware features of the iSCSI SAN architecture on System i5 are:

- ▶ The iSCSI network is a switched 1 Gb Ethernet network. Whether you are connecting xSeries server or BladeCenter to System i5, all iSCSI connections must go through a switch. In the case of xSeries, this must be an external 1 Gb switch as shown in Figure 2-4 on page 13. In the case of BladeCenter, the switch can be a 1 Gb switch module (or modules) in the BladeCenter itself, as shown in Figure 2-4 on page 13.
- ▶ Only IBM xSeries servers and BladeCenters are supported. For a list of supported models, go to this Web site:
<http://www-03.ibm.com/systems/i/bladecenter/iscsi/servermodels/>
- ▶ xSeries servers require at least one IBM iSCSI network adapter.
- ▶ Each Blade server requires an IBM iSCSI network adapter.
- ▶ One or more iSCSI network adapters are required in the System i5.

For detailed information about iSCSI hardware requirements, refer to Chapter 3, “Planning for iSCSI attached servers” on page 43.

2.2.2 iSCSI hardware terminology

In the same way that TCP/IP is used to transport data over a network, SCSI is a protocol that computer systems use to access data stored on disk, tape, and optical drives. The SCSI protocol provides high performance data access and is widely used in the industry. iSCSI is the marriage of these two ubiquitous protocols into a single protocol which enables an Ethernet network to transport the SCSI protocol in TCP/IP packets. iSCSI provides tremendous advantages in the flexibility it provides to transport storage-type data over a standard Ethernet network. For this reason, the iSCSI standard was chosen as the basis for implementing the next generation of System i5 SAN.

The following list includes basic iSCSI terminology that you need to know in order to understand the various components of the System i5 iSCSI network:

- ▶ **SCSI:** Small Computer System Interface
A protocol used by computers to access data stored on disk, tape, and optical drives.
- ▶ **iSCSI:** Internet SCSI (Small Computer System Interface)
SCSI commands and data sent across a network in TCP/IP packets. It was developed as a storage area networking standard for linking data storage facilities.
- ▶ **HBA:** Host Bus Adapter
A physical hardware interface used to connect a computer to a network for the purpose of sending and receiving storage data.
- ▶ **Target HBA:** The HBA adapter that is receiving iSCSI requests for data. (In our case, this is the System i5 end of the iSCSI connection.)
Target HBAs are also referred to as *storage paths*.
- ▶ **Initiator HBA:** The HBA adapter that is sending iSCSI requests for data. (In our case, this is the hosted server, xSeries or Blade server).

2.3 iSCSI network basics

iSCSI attached servers are standard xSeries or BladeCenter server models that have processors, memory, and expansion cards, but no disks. All of the disk space is in the iSeries server and managed in the same way that IXS server and IXA server disk space is managed. The installation procedure for an iSCSI attached Windows server requires hardware to be installed and configured in the iSeries, xSeries, or BladeCenter servers. As with the IXA, the iSCSI HBA attached xSeries servers have their own expansion slots, so additional options can be installed to expand the capabilities of the server. Figure 2-4 illustrates a typical System i5 iSCSI network.

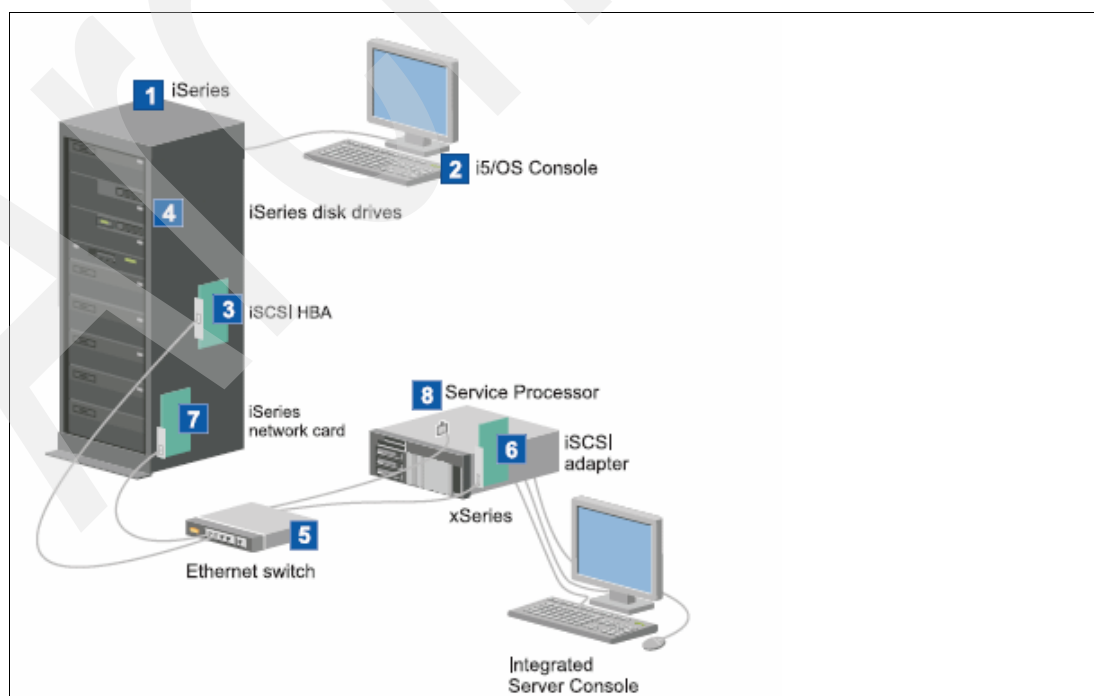


Figure 2-4 Basic iSCSI connectivity to System i5

In Figure 2-4 on page 13, you see the following:

1. POWER5-based iSeries and System i5 models are the only ones that support iSCSI connectivity.
2. The i5/OS console, which can be an HMC, a PC running Operations console, or even a 5250 green screen.
3. Depending on the type of the physical network you are connecting to, copper or fiber iSCSI HBAs are available. This iSCSI adapter serves as the target HBA and connects to a 1 Gb switched Ethernet network using standard Ethernet cabling.
4. A hosted xSeries or Blade server does not have its own hard disk drives. i5/OS provides logical disk drives, taken out of single level storage for the hosted server to use. These logical drives, and other iSeries storage devices, are accessed through the iSCSI HBA.
5. The iSCSI HBA network cables are connected to a standard Gigabit Ethernet switch.

Note: You must have a switch to connect the HBAs that comprise the iSCSI network. At the time of writing, direct connection between an iSCSI HBA in the System i5 and an iSCSI HBA in a hosted xSeries or Blade server is not supported. Furthermore at the time of writing, the iSCSI network does not support TCP/IP routing.

6. An iSCSI HBA is required in the xSeries or Blade server. This adapter provides the connection to the iSCSI HBA for iSeries. This iSCSI adapter serves as the initiator HBA and also connects to the 1 Gb switched Ethernet network using standard Ethernet cabling. The xSeries or Blade server sees there is an HBA as the storage adapter where the disks are to be found. If there are more than one initiator HBA in the hosted server, one is nominated as the boot device.
7. An System i5 Ethernet LAN connection is required by IBM Director to discover and manage remote xSeries or BladeCenter servers across the network. IBM Director connects to the Service Processor on an xSeries server or the Management Module on a BladeCenter. The System i5 LAN adapter can be connected to the 1 Gb iSCSI LAN, but it is usually connected to a separate Ethernet LAN.
8. The Service Processor in an xSeries server, or Management Module in a BladeCenter, allows the System i5 to discover and manage the hosted system. The service processor can be a Remote Supervisor Adapter (RSA II), Baseboard Management Controller (BMC), or a Management Module in a BladeCenter. The RSA II, BMC, or Management Module can be connected to the System i5 over the 1 Gb iSCSI LAN, but it is usually connected to a separate Ethernet LAN.

Note that the use of the iSCSI adapter as a general purpose external network connection is not supported.

2.3.1 Sending SCSI and VE LAN data over an iSCSI network

As shown in Figure 2-5 on page 15, each target and initiator HBA in the System i5 iSCSI network can support two types of connection:

- ▶ A SCSI connection for the transmission of storage data using the SCSI protocol over TCP/IP (iSCSI)
- ▶ A Virtual Ethernet LAN connection for the transmission of LAN data using TCP/IP

Think of the iSCSI connection between a target and initiator HBA as a “pipe” which is divided into two more pipes; one for carrying storage data and one for carrying LAN data.

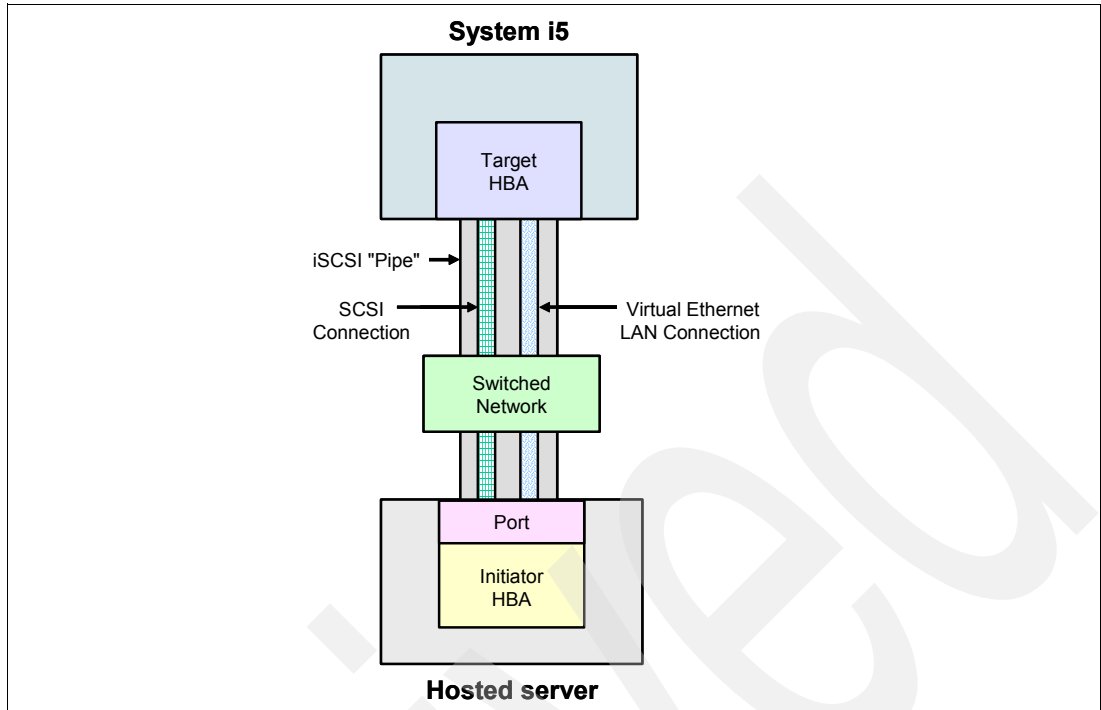


Figure 2-5 Running SCSI and VE LAN connections over the same iSCSI pipe

The configuration of the target and initiator HBAs in an iSCSI network is very flexible. For example, if there are two target HBAs communicating with two initiator HBA ports, both connections can carry both storage and LAN data as shown in Figure 2-5, or you can dedicate one connection to SCSI data and the other connection to LAN data, as shown in Figure 2-6 for an xSeries server.

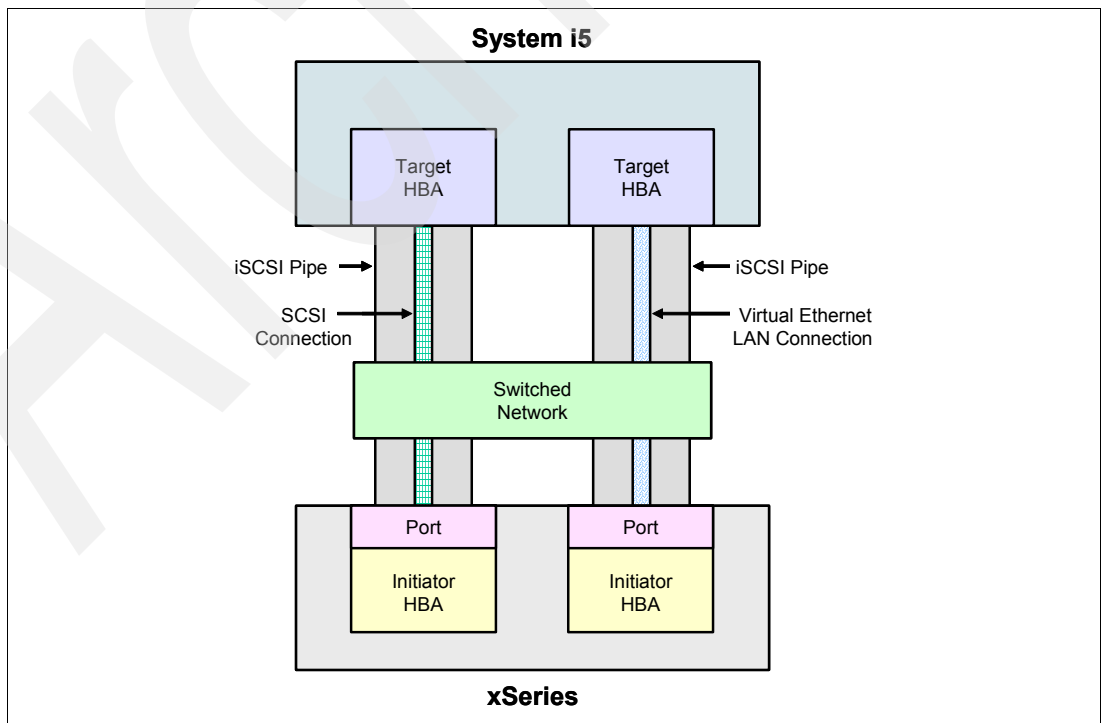


Figure 2-6 Running SCSI and VE LAN connections over different iSCSI pipes: xSeries server

The situation is similar, but also a little different in the case of a Blade server. As shown in Figure 2-7, you can dedicate a target-initiator HBA port connection to SCSI or LAN data, just as for an xSeries server. The difference is that on a Blade server, the initiator HBA supports two logical ports rather than just one as on the xSeries server initiator HBA.

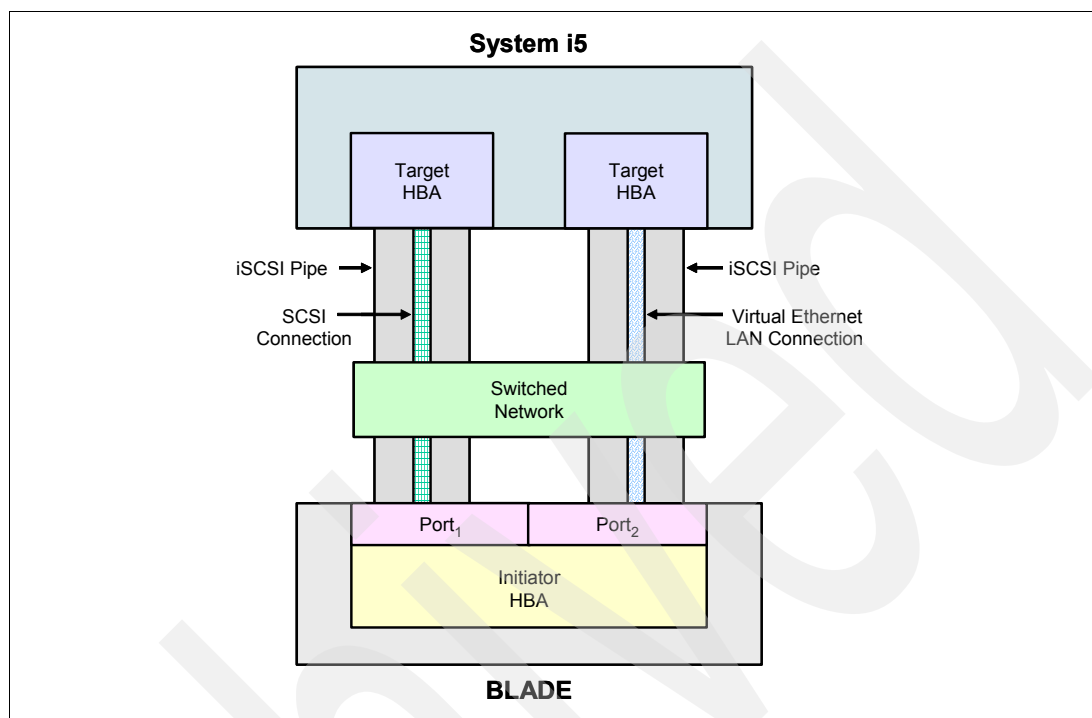


Figure 2-7 Running SCSI and VE LAN connections over different iSCSI pipes: Blade server

2.3.2 Comparing targets, initiators, and ports

iSCSI connections are made between ports on the target and initiator HBAs. We do not talk about target HBA ports, because there is always only one port on a target HBA. However, on an initiator HBA, we do talk about ports because ports are significant. On an initiator HBA for an xSeries server, there is only ever one port. However, on an initiator HBA for a Blade server, there are either one or two logical ports available, even though there is only one physical HBA installed on the Blade server. To access the first port on the initiator HBA in a Blade server, you must install a switch module in bay 3. To access the second port on the initiator HBA in a Blade server, you must install a switch module in bay 4. To access both ports on the initiator HBA in a Blade server, you must install a switch module in both bay 3 and bay 4.

You can have up to four initiator HBAs installed in an xSeries server, providing you with up to four ports. You can only install one initiator HBA in a Blade server, providing you with up to two ports, depending on whether you have one or two switch modules installed in bay 3 and bay 4 as previously described.

When you have multiple initiator HBA ports available in an xSeries or Blade server, one (and only one) of the initiator HBA ports must be identified as the boot port in the Remote system configuration object (RMTSYS). The boot port is where the hosted server looks for a bootable drive, or in the case of i5/OS, a storage space containing the operating system. If there is only one initiator HBA port available in the hosted server, this port is automatically defined as the boot port. In this case you do not need to do anything.

The initiator HBA where the boot port is defined is known as the *boot device*.

2.3.3 The basic iSCSI configuration

i5/OS represents iSCSI attached xSeries and IBM BladeCenter servers similar to the way it represents IXS attached and IXA attached xSeries servers. However, the iSCSI architecture requires additional i5/OS objects and configuration information that were not required for IXS attached and IXA attached xSeries servers. Because iSCSI attached servers are connected to System i5 using an Ethernet network (rather than the system bus/HSL attachment that is used with the IXS and IXA), additional configuration information is required to identify and communicate with the xSeries or Blade server on the iSCSI network. In addition, because iSCSI attached servers can coexist with other systems on the Ethernet network, security of communications and data flows between i5/OS and the iSCSI attached servers is important.

A basic iSCSI configuration consists of an iSCSI target HBA installed in a System i5 i5/OS partition and an iSCSI initiator HBA installed in an xSeries or Blade server. This “basic iSCSI configuration” is what you set up when you work through Chapter 3, “Planning for iSCSI attached servers” on page 43 and Chapter 4, “Installing the iSCSI integrated server” on page 53 in this redbook. The concept of a basic iSCSI configuration is important, because it is the basis upon which we build more complex configurations as described in Chapter 9, “Scaling your iSCSI network” on page 325.

An example of a basic iSCSI connection between an i5/OS partition and a hosted xSeries server is shown in Figure 2-8.

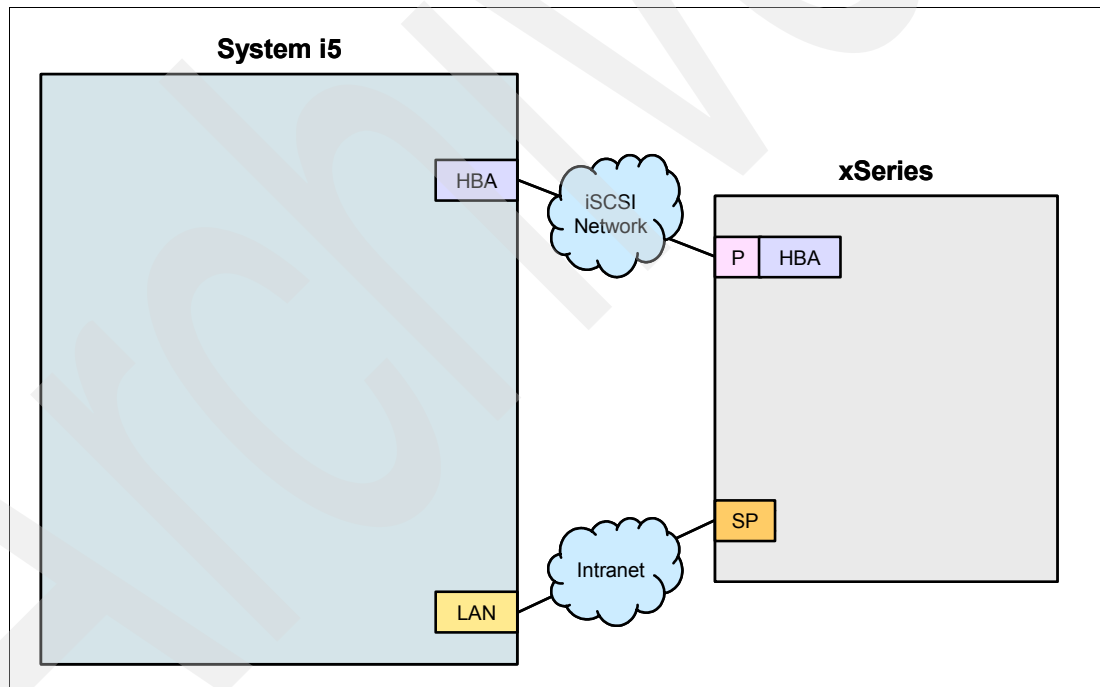


Figure 2-8 Basic iSCSI configuration: xSeries

Figure 2-8 shows a connection between a single target HBA in the i5/OS hosting partition and a single initiator HBA port in an xSeries server across an iSCSI network.

The equivalent of Figure 2-8 for a Blade server is shown in Figure 2-9 on page 18.

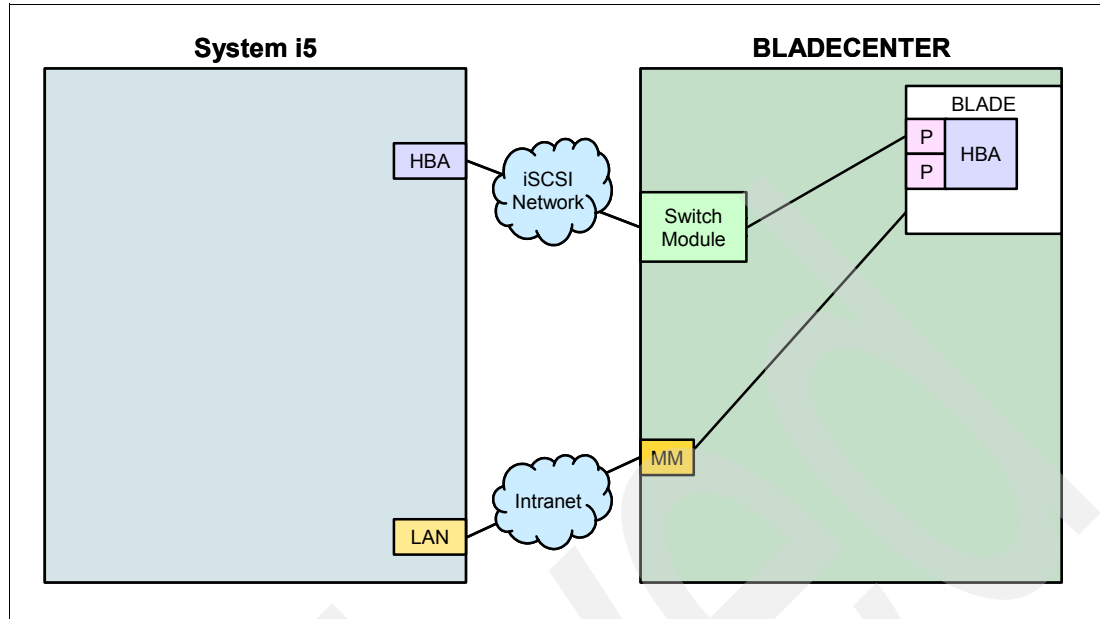


Figure 2-9 Basic iSCSI configuration: Blade server

Note that in Figure 2-9, only one initiator HBA port is active because there is only one switch module installed in either bay 3 or 4. Therefore, the second initiator HBA port is not available.

The basic iSCSI configurations shown here are the ones that you create when you work through Chapter 3, “Planning for iSCSI attached servers” on page 43 and Chapter 4, “Installing the iSCSI integrated server” on page 53, as described in Chapter 9, “Scaling your iSCSI network” on page 325.

It also shows a connection between an Ethernet LAN adapter in the i5/OS hosting partition and the service processor in an xSeries server. Note that the target HBA and LAN adapter shown in Figure 2-8 on page 17 must be in the same i5/OS partition.

Two distinct networks are illustrated in Figure 2-8 on page 17 and Figure 2-9.

► **iSCSI network**

The target and initiator HBAs are connected over a 1 Gb Ethernet LAN, which uses an isolated switch. The switch should be isolated physically for security reasons, and there should be no nodes on the network other than target and initiator HBAs for reasons of security, performance, and availability. To secure the connections on the iSCSI network, we recommend that you use the CHAP protocol, as described in 2.7, “iSCSI security model” on page 30.

► **Intranet**

The service processor connection uses an external Ethernet network, which would normally be the site LAN or intranet, although it could also be a wide area network. This enables you to access the service processor from anywhere on your intranet or wide area network to manage the hosted server. The service processor connection *could* use the same isolated switch as the iSCSI network; however, the i5/OS LAN adapter would not be available for communication on the intranet. We therefore recommend that you connect it to your site LAN, and you secure the connection using user ID and password security as described in 2.7, “iSCSI security model” on page 30.

2.4 Objects that define the iSCSI network

Here we describe the i5/OS objects that comprise the iSCSI network.

First we look at the i5/OS objects that you need to create in order to install the basic iSCSI configuration. We then compare this with a scenario where we have multiple hosted servers and describe the additional objects that we need to create.

2.4.1 Objects for a single hosted server

In Figure 2-10, we see the i5/OS objects that you need to configure in order to install the basic iSCSI configurations shown in Figure 2-8 on page 17 and Figure 2-9 on page 18.

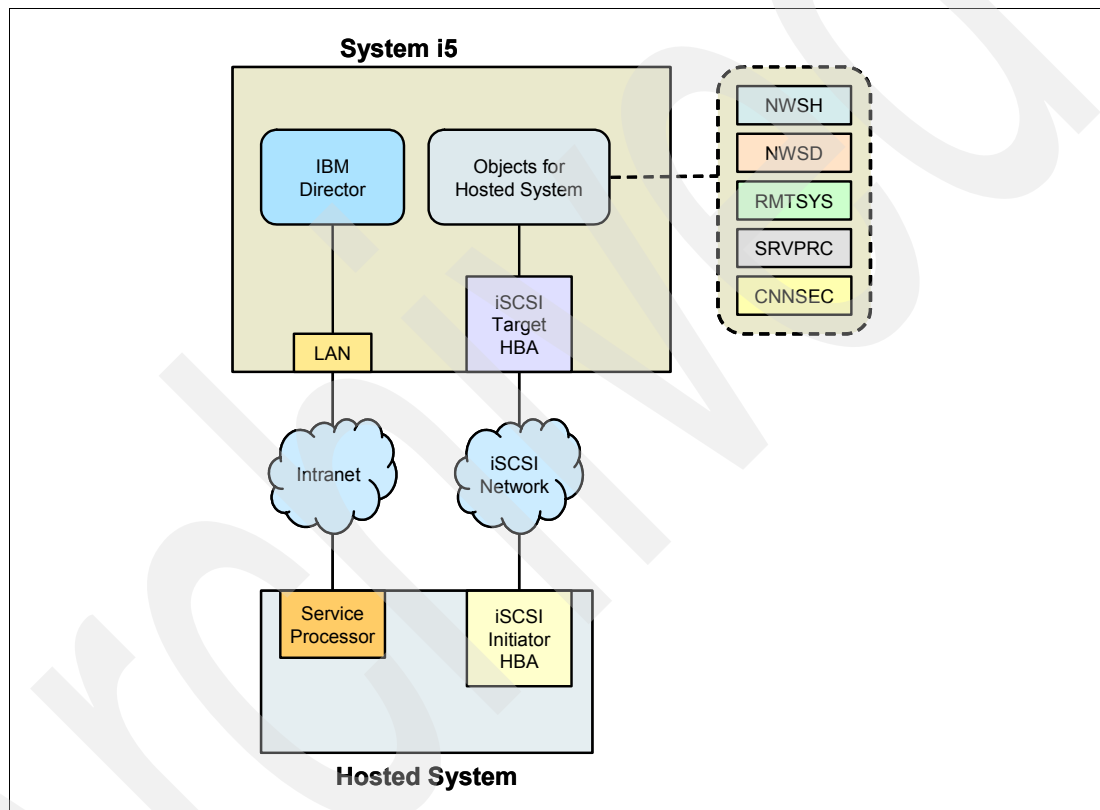


Figure 2-10 Objects for a single hosted server

Figure 2-10 shows that there are five different i5/OS objects that describe a basic iSCSI connection between an i5/OS hosting partition and a hosted xSeries or Blade server. They are:

- ▶ **NWSH** - Network server host adapter
This object represents a target HBA.
- ▶ **NWSD** - Network server description
This object describes the i5/OS environment created for the hosted server.
- ▶ **RMTSYS** - Remote system configuration
This object describes the hosted xSeries or Blade server.
- ▶ **SRVPRC** - Service processor configuration

This object describes the Service Processor configuration (xSeries) or Management Module configuration (BladeCenter).

- CNNSEC - Communications security configuration

This object describes the security configuration for the iSCSI connection.

Note that i5/OS locates and manages hosted systems by sending commands to the service processor of the remote system over an Ethernet network via a LAN adapter installed in the i5/OS hosting partition. IBM Director is used for these functions and must be installed and running in the partition that is connected to the iSCSI network.

The relationship between the objects for the basic iSCSI configuration shown in Figure 2-10 on page 19 can be represented by Figure 2-11.

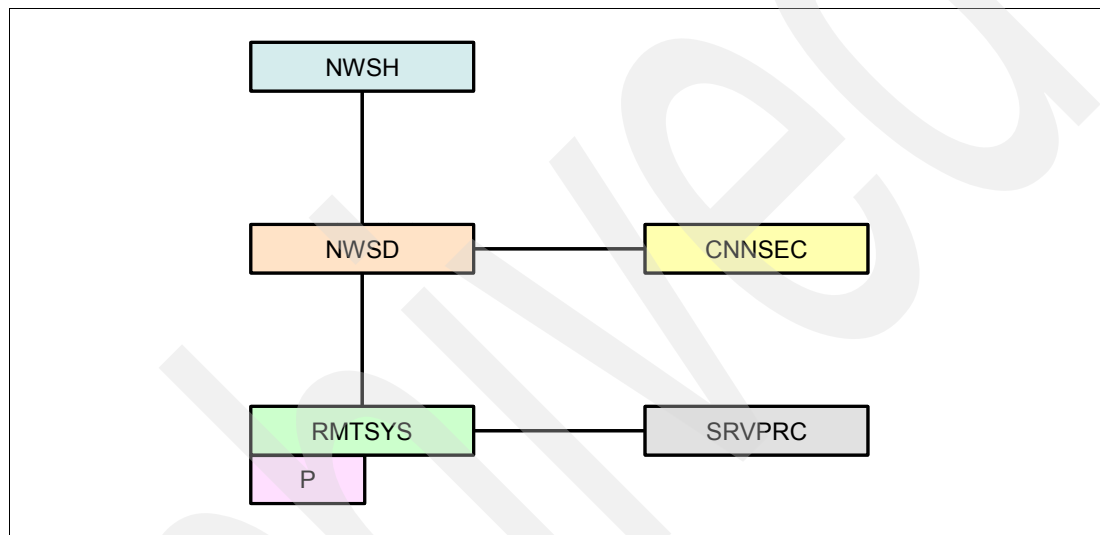


Figure 2-11 Object relationships for a single hosted server - 1

Figure 2-11 is a very important diagram, because it demonstrates the fundamental relationships between the objects that make up a hosted server connected to an i5/OS partition over an iSCSI network. We use variations of this diagram to describe the more complex configurations covered in Chapter 9, “Scaling your iSCSI network” on page 325.

You should memorize Figure 2-11.

Note the following important points shown in Figure 2-11:

- An NWSD can be linked to only one RMTSYS object, and vice versa. There is always a 1:1 relationship between the NWSD and RMTSYS objects.
- In the case of an xSeries server, a RMTSYS object can be linked to only one SRVPRC object, and vice versa. There is always a 1:1 relationship between the RMTSYS and SRVPRC objects.
- In the case of a Blade server, a RMTSYS object can be linked to only one SRVPRC object, but an SRVPRC object can be linked to multiple RMTSYSs. This is because you have one Management Module in a BladeCenter to service up to 14 Blade servers.
- The “P” attached to the RMTSYS object in Figure 2-11 represents the iSCSI boot port on an initiator HBA. The boot port is not a separate object; it is specified in the RMTSYS object. However, we show the boot port, because it is important when we begin to describe the various ways you can scale the System i5 iSCSI network.

- The CNNSEC object is planned to provide IPsec security for the iSCSI network, but this capability was not implemented at the time of writing. However, you still need to configure the CNNSEC object. Note that you can connect a single CNNSEC object to all your NWSDs. Therefore, you only need one for your entire i5/OS partition. When the IPsec capability becomes available, there might be reasons why you would configure multiple CNNSEC objects, but until then, there is no reason to configure more than one.

Adding target HBA connections to a hosted server

Figure 2-12 shows how you can have multiple target HBAs (NWSHs) connected to a single hosted server (NWSD).

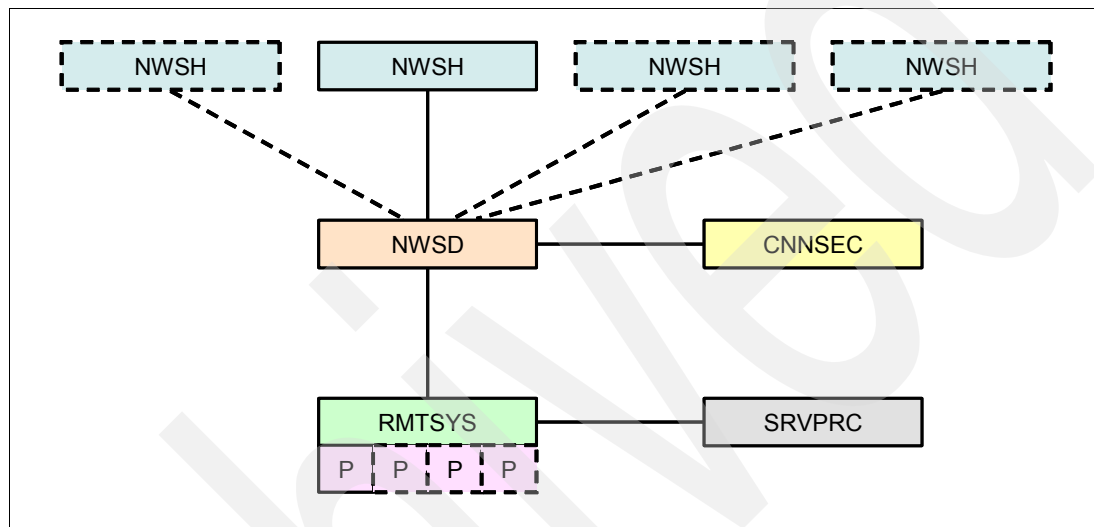


Figure 2-12 Object relationships for a single hosted server

Note the following important points shown in Figure 2-12:

- You can add up to four target HBA connections to a single hosted xSeries server. You would also probably want to add additional initiator HBA ports, designated by “P” in Figure 2-12 to balance the bandwidth between the i5/OS partition and hosted server connections.
- You can add up to four initiator HBA ports to a hosted xSeries server.

However, there is a rule you must remember when planning your iSCSI network.

Important: The number of target HBAs connected to the hosted server must be equal to or greater than the number of initiator HBA ports in the hosted server.

In other words, you can never have more initiator ports active in a hosted server than you have target HBAs (storage paths) configured for that hosted server.

Figure 2-13 on page 22 is the same as Figure 2-12, but for a Blade server. The same comments apply, except that you can have a maximum of only two initiator HBA ports in a Blade server.

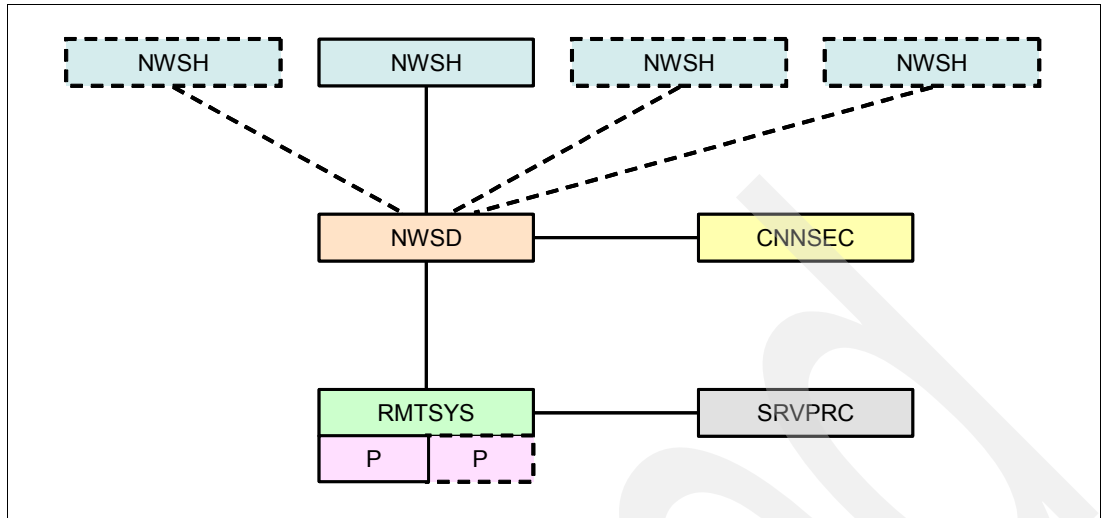


Figure 2-13 Object relationships for multiple hosted servers: Blade server

2.4.2 Objects for multiple hosted servers

You can scale your iSCSI network by connecting additional hosted servers. Although a single target HBA can host multiple xSeries or Blade servers, to provide adequate bandwidth for these extra servers you would probably want to add additional target HBAs. An example of multiple server support is shown in Figure 2-14.

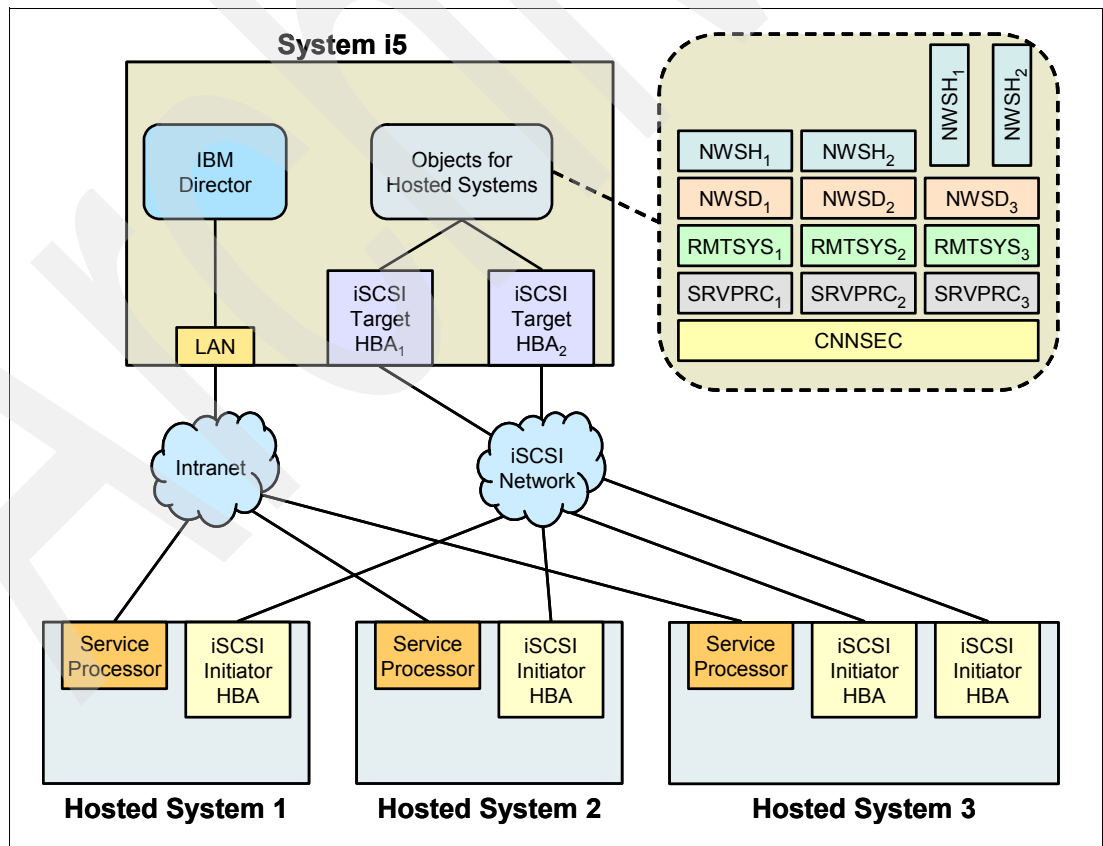


Figure 2-14 Objects for multiple hosted servers

In Figure 2-14 on page 22, you can see that two target HBAs are shared among three hosted systems. Note the following important points shown in Figure 2-14 on page 22:

- ▶ Each hosted system requires its own set of i5/OS objects.
- ▶ A target HBA can establish connections to one or more (up to eight) hosted systems.
- ▶ Hosted system 1 has a connection to target HBA 1 (NWSH 1).
- ▶ Hosted system 2 has a connection to target HBA 2 (NWSH 2).
- ▶ Hosted system 3 has a connection to both HBA 1 (NWSH 1) and HBA 2 (NWSH 2).
- ▶ All three hosted systems are sharing a single communications security object (CNNSEC).

For detailed information about scaling of the System i5 iSCSI environment, refer to Chapter 9, “Scaling your iSCSI network” on page 325.

2.5 iSCSI object descriptions

The following sections discuss the iSCSI object descriptions related the System i5 configuration.

2.5.1 Network server description (NWSD)

The network server description (NWSD) is the key i5/OS configuration object for all types of integrated servers. The NWSD object is used to tie together all of the other i5/OS objects that relate to an integrated server. For example, it contains a reference to the hardware that the server runs on, links to the virtual disk drives that the server uses, references to the network ports that the server uses and many other attributes of the server. The i5/OS Install Windows Server (INSWNTSVR) command is used to create the server's NWSD and several other i5/OS objects that are needed by the server. For a description of the values that the NWSD contains, see the i5/OS Create Network Server Description (CRTNWSD) command. For an integrated server, the IXS attached and IXA attached xSeries server hardware is controlled by i5/OS. An integrated server is started by varying on the NWSD for that server. This initiates the Windows operating system boot process. An integrated server is shut down by varying off the NWSD for that server. This initiates the Windows operating system shutdown process. For an IXS, i5/OS communicates directly with the IXS hardware to perform the start and shut down tasks. For an IXA attached xSeries server, i5/OS communicates over a high speed link (HSL) bus with the IXA that is installed in the xSeries server to initiate the start and shut down tasks. The IXA in turn communicates with the service processor (SP) of the xSeries system to perform the start and shut down tasks.

Note: Because the IXA provides a hard-wired connection to the xSeries service processor, an i5/OS object is not needed to configure the xSeries service processor characteristics.

The network server description (NWSD) object is basically the same as described for Figure 2-11 on page 20, except for the following:

- ▶ It contains a reference to a remote system configuration object instead of an iSeries hardware resource name.
- ▶ Unlike an IXA attached server, which uses one IXA card in the xSeries system to manage all of the SCSI and Virtual Ethernet data flows, on an iSCSI attached server solution both the iSeries and the xSeries can have multiple iSCSI host bus adapters (HBAs). This allows multiple SCSI and Virtual Ethernet data paths between the iSeries and xSeries or IBM BladeCenter systems, which can provide greater bandwidth and connection redundancy.
- ▶ You can define one or more storage paths. These storage paths reference the NWSH objects that are associated with the iSCSI HBAs that are used by the integrated server.

You can choose which storage path is used for the SCSI data flows for each virtual disk drive. By associating your virtual disk drives with different storage paths, you can spread the overall server SCSI data flow workload across the storage path iSCSI HBAs for greater bandwidth.

- ▶ You can define a multi-path group, which is a subset of the configured storage paths. You can then associate a virtual disk drive with the multi-path group, instead of associating it with a specific storage path. Using the multi-path group for a virtual disk drive has the advantage that if the iSCSI HBA for one of the NWSHs in the multi-path group fails or the network connection to the iSCSI HBA fails, the SCSI data flow workload for that virtual disk drive is automatically routed to one of the other iSCSI HBAs that is configured in the multi-path group. This provides connection redundancy and improves availability.
- ▶ You can define one or more Virtual Ethernet paths. These Virtual Ethernet paths also reference the NWSH objects that are used by the integrated server. You can choose which NWSH is used for each Virtual Ethernet port that the integrated server uses. By associating different Virtual Ethernet ports with different NWSHs, you can spread the overall server Virtual Ethernet data flow workload across the Virtual Ethernet path iSCSI HBAs for greater bandwidth.
- ▶ Just as with an IXS attached or IXA attached server, the iSCSI attached xSeries or IBM BladeCenter server hardware is controlled by i5/OS.
 - An iSCSI attached server is started and shut down the same as with an IXS attached or IXA attached server by varying on or off the NWSD for that server. For more information, see Chapter 6, “Managing integrated iSCSI environments” on page 149.
 - For an iSCSI attached xSeries or IBM BladeCenter server, i5/OS communicates over an Ethernet network with the service processor (SP) for the xSeries system or the IBM BladeCenter management module for an IBM BladeCenter server to perform the start and shut down tasks.

For server hardware power control, the major difference between the IXS/IXA configurations and the iSCSI configuration is that for IXS attached or IXA attached servers, the server hardware is identified by the iSeries hardware resource name, while for iSCSI attached servers, the server hardware is identified by the remote system configuration object.

2.5.2 Network server host adapter

The network server host adapter (NWSH) device description object shown in Figure 2-11 on page 20 represents the iSCSI host bus adapter (HBA) that is used by the iSeries side of the iSCSI connection:

- ▶ It identifies the iSeries hardware resource name (for example, LIN33) for the iSCSI HBA.
- ▶ It defines how communications errors are logged and communications recovery information.
- ▶ It defines the Internet addresses, ports, and so on for the SCSI and LAN interfaces on the iSCSI HBA. The iSeries can have multiple iSCSI HBAs, each with an associated NWSH object.
- ▶ Each NWSH can be shared by multiple integrated servers. In configurations where bandwidth is not a concern, this results in a lower cost solution.
- ▶ Each integrated server can use multiple NWSHs. This allows multiple SCSI and Virtual Ethernet data paths between the iSeries and the xSeries or IBM BladeCenter systems, which can provide greater bandwidth and connection redundancy.

FSDxx and CMNxx resource usage

There are limits to the number of storage and Virtual Ethernet paths that an iSCSI HBA can support. Each active server storage path will use a file server resource in the network server host adapter (NWSH) object that corresponds to the iSCSI HBA. Likewise, each active server Virtual Ethernet path will use a Virtual Ethernet resource in the NWSH object. There is a limit to the number of file server and Virtual Ethernet resources supported by a particular NWSH, which limits how many active servers can use the NWSH. To see the NWSH file server and Virtual Ethernet resource limits using iSeries Navigator, follow these steps:

1. Expand Integrated Server Administration.
2. Expand iSCSI Connections.
3. Select Local Host Adapters.
4. Right-click a NWSH from the list available.
5. Select **Properties**.
6. Click the Resource Usage tab. You see a window similar to that shown in Figure 2-15.

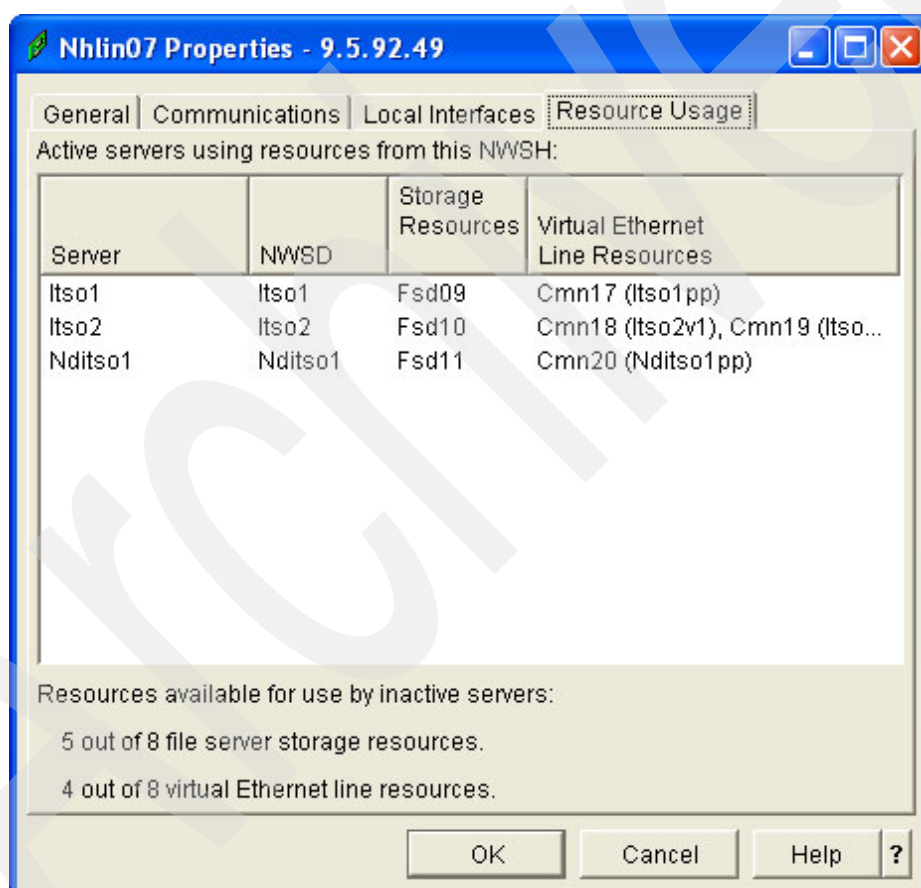


Figure 2-15 Hosted server and Virtual Ethernet resource usage on an NWSH

The table shows the active servers that are currently using the NWSH and the file server and Virtual Ethernet resources that they are currently using. Below the table, it shows how many file server and Virtual Ethernet resources are still available for use by inactive servers and the total number of file server and Virtual Ethernet resources that the NWSH supports, which is 8. Click **Cancel** on the NWSH properties panel to close the panel. If you want to use a CL command, see the WRKDEV or DSPDEV commands. There is also a less defined practical limit to the number of servers that an iSCSI HBA can support. The practical limit is

determined by the available iSCSI HBA bandwidth and the workload that is being run through the iSCSI HBA. The practical limit will most likely limit how many hosted systems the iSCSI HBA can support before the file server and Virtual Ethernet resource limits described above are reached. The practical limits depend on your particular server configurations and workloads.

To locate a target HBA, follow these steps:

1. Expand **Configuration and Service**.
2. Select **Communications**.
3. Scroll down until you see a resource of type Network Server Host Adapter.

Notice that there are eight Fsdxx resources for NWSDs and eight Cmnxx resources for Virtual Ethernet LANs.

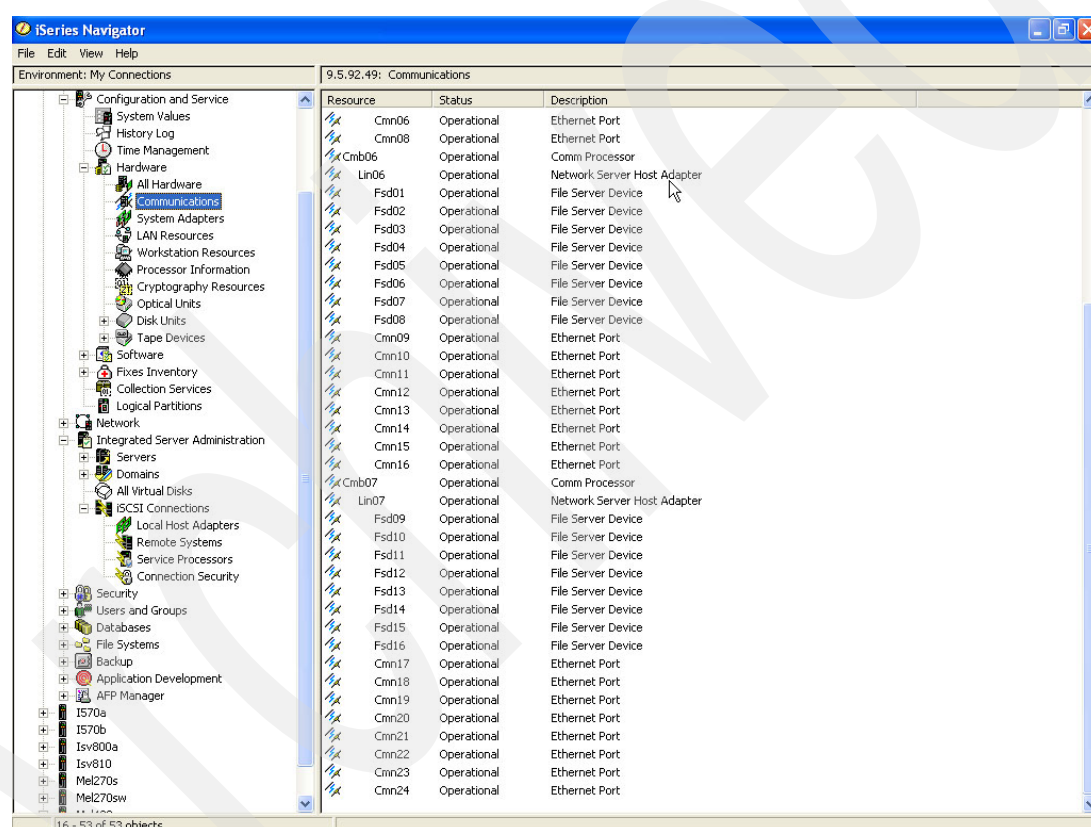


Figure 2-16 Target HBA: NWSD slots and Virtual Ethernet LAN resources

Storage paths

In i5/OS configuration objects, network interface information is labeled as local or remote. These terms are relative to i5/OS. Local interface information is for the i5/OS side. Remote interface information is for the Windows hosted system side.

The network server host adapter and the remote system configuration define IP address information for opposite sides of the iSCSI network. When connected by a simple, switched network, the following rules apply:

- ▶ The SCSI Internet addresses in these two objects that are connected by a switch must be in the same subnet. For example, with IP addresses of the form a.b.x.y and 255.255.255.0 subnet masks, a.b.x must be the same value for both objects.
- ▶ The LAN Internet addresses in these two objects that are connected by a switch must be in the same subnet.
- ▶ In the network server host adapter, the gateway elements can be any unassigned IP address in any subnet if you do not have a gateway in your network.
- ▶ In the remote system configuration, the gateway elements should be blank if you do not have a gateway in your network.

2.5.3 Remote system configuration

A remote system network server configuration (NWSCFG type RMTSYS) object (shown in Figure 2-11 on page 20) represents the iSCSI attached xSeries or IBM BladeCenter server:

- ▶ It identifies the server hardware by serial number, type, and model.
- ▶ It contains configuration information for the iSCSI host bus adapters (HBAs) that are used by the xSeries or IBM BladeCenter server.
- ▶ It contains values required to boot the server (such as specifying which iSCSI adapter to boot from).
- ▶ It contains a reference to the service processor NWSCFG object (see below) that is used to control the xSeries or IBM BladeCenter server.
- ▶ The remote system configuration can optionally contain values used to secure the server boot process. The xSeries or IBM BladeCenter server can have multiple iSCSI HBAs. This allows multiple SCSI and Virtual Ethernet data paths between the iSeries and the xSeries or IBM BladeCenter systems, which can provide greater bandwidth and connection redundancy. The remote system configuration object for an integrated server is referenced via a parameter in the NWSD.

Boot Port

All of the iSCSI attached stand-alone or IBM BladeCenter servers are diskless and require an xSeries or IBM BladeCenter iSCSI host bus adapter (HBA) as a boot device. Both the i5/OS remote system configuration and the remote server iSCSI HBA must be configured before you install or use a new integrated Windows server.

The iSCSI HBA must be configured during the xSeries or IBM BladeCenter boot process using the adapter CTRL-Q utility. We recommend that it is configured as part of the initial server setup. There is a minimal set of required parameters for you to configure in the hosted server iSCSI HBA. These parameters need to be matched to those parameters configured in the remote system configuration object. The parameters vary depending on the selected boot mode. See the iSCSI install readme first Web page for details about how to configure the hosted system iSCSI HBA as the iSCSI boot device.

Enabling the hosted server boot device

The iSCSI HBA installed in an xSeries or IBM BladeCenter acts as a boot device during the boot process, based on the configured parameters. When the xSeries has only one iSCSI HBA, this adapter needs to be configured as the boot device. iSCSI boot is enabled by default on all iSCSI HBAs, but you must configure additional information. When the xSeries server has multiple iSCSI HBAs installed, only one of these must be configured as the boot device.

The IBM BladeCenter server iSCSI HBA is a dual port adapter. Only one port is required to be configured as a boot device.

One of the HBA's ports must be identified as the boot port. The boot port is the HBA where the hosted server will look for a bootable disk drive, or in the case of i5/OS, a storage space containing the operating system. For example, in Figure 2-14 on page 22, P₁ has been specified as the boot drive in the RMTSYS configuration object. If there is only one port in the hosted server, this port is automatically defined as the boot port.

2.5.4 Service processor configuration

A service processor network server configuration (NWSCFG type SRVPRC) object (shown in Figure 2-11 on page 20) represents the xSeries service processor or the IBM BladeCenter management module:

- ▶ It identifies the service processor or management module hardware by serial number and type and model.
- ▶ It defines how to find the service processor or management module on the Ethernet network using an Internet address or host name.
- ▶ The service processor object can optionally contain values used to secure the i5/OS to service processor communications.

Note: For iSCSI attached xSeries servers, there is a one-to-one relationship between the service processor object and the remote system configuration, because each service processor controls only one xSeries server. However for iSCSI attached IBM BladeCenter servers, there can be a one-to-many relationship between the service processor object and the remote system configuration, because each management module can control any of the IBM BladeCenter servers that are contained within the IBM BladeCenter chassis. Therefore with iSCSI attached IBM BladeCenter servers, it would be common for several remote system configurations to share (refer to) the same service processor object.

Remote Service Processor (SRVPRC) defines the connection to the BladeCenter's Management Module (service processor). The SP interface is used to power the Blade servers on or off. This is controlled via the Management Module (MM). The Management Module functions as a service processor and a keyboard/video/mouse (KVM) for all of the Blade servers installed in a BladeCenter unit. It controls the external keyboard, mouse, and video connections, for use by a local console, and a 10/100 Mbps Ethernet remote management connection. You will have one of these defined for the MM to be used for all of the Blades in the BladeCenter.

This physical connection is required so that the hosting i5/OS can communicate with the service processor of the hosted system. The connection can consist of a simple, switched network or a more complex, routed network. Windows environment on iSeries uses IBM Director over this connection to manage the state of the hosted system. At one end of the connection is a LAN adapter or adapters controlled by i5/OS. This LAN adapter can still be available for other uses. The IP address and other attributes of this adapter are controlled using standard i5/OS configuration methods. Windows environment on iSeries does not configure this adapter. It can automatically discover the service processor using IBM Director and one or more i5/OS TCP interfaces that are already configured. At the other end of the connection is the service processor. The service processor has its own Ethernet port and TCP/IP stack. This TCP/IP stack is active whenever the server's power cord is plugged into

an energized AC outlet, even if the server is not in a powered on state. On certain xSeries models, a single Ethernet port can be shared by Windows and a particular type of service processor, known as the Baseboard Management Controller (BMC). In this case, the same physical port on the hosted system provides both the service processor connection and an external network connection.

This physical connection is required so that the hosting i5/OS can communicate with the service processor of the hosted system. The connection can consist of a simple, switched network or a more complex, routed network. Windows environment on iSeries uses IBM Director over this connection to manage the state of the hosted system. At one end of the connection is a LAN adapter or adapters controlled by i5/OS. This LAN adapter can still be available for other uses. The IP address and other attributes of this adapter are controlled using standard i5/OS configuration methods. Windows environment on iSeries does not configure this adapter. It can automatically discover the service processor using IBM Director and one or more i5/OS TCP interfaces that are already configured. At the other end of the connection is the service processor. The service processor has its own Ethernet port and TCP/IP stack. This TCP/IP stack is active whenever the server's power cord is plugged into an energized AC outlet, even if the server is not in a powered on state. On certain xSeries models, a single Ethernet port can be shared by Windows and a particular type of service processor, known as the Baseboard Management Controller (BMC). In this case, the same physical port on the hosted system provides both the service processor connection and an external network connection.

2.5.5 The role of IBM Director

i5/OS locates and manages hosted systems by sending commands to the service processor of the remote system over an Ethernet network via a LAN adapter installed in the i5/OS hosting partition. IBM Director is used for these functions and must be installed and running in the partition that is connected to the iSCSI network.

2.6 Other components of the iSCSI network

Here is a detailed description of the other components of the iSCSI network.

2.6.1 Network server storage spaces

A network server storage space (NWSSTG and called a storage space) represents a virtual disk drive that the server uses. Virtual disk drives can vary in size from 1 MB to 1000 GB each. Up to 64 virtual disk drives can be linked to a server, depending on the server configuration, so the storage capacity of an integrated server can range from several gigabytes to many terabytes. The virtual disk drives are first created as stand-alone objects and then linked to the integrated server by identifying the NWSD of the integrated server that uses them.

Each server has at least two virtual disk drives that are automatically created by the INSWNTSVR command but can also have user-defined virtual disk drives.

The drives are:

- ▶ The system drive (typically the C: drive) contains the Windows server operating system (such as Windows Server 2003).
- ▶ The install drive (typically the D: drive) contains a copy of the Windows server installation media as well as the portion of the i5/OS Integrated Server Support (product 5722-SS1 option 29) code that runs on the Windows server. The install drive is used during the

Windows installation process and is also used every time the server is started to pass configuration information from i5/OS to the server.

- ▶ Additional user-defined drives are typically used for server applications and data.

The actual disk storage for the virtual disk drives is allocated from the i5/OS integrated file system (IFS). The virtual disk drives can be allocated from the default system disk pool (also known as the *system auxiliary storage pool*, or *system ASP*) or from a user-defined disk pool or an independent disk pool (IASP).

Notes:

1. Because virtual disk drives are objects in the i5/OS IFS, an entire virtual disk drive image can be backed up and restored using the i5/OS Save (SAV) and Restore (RST) commands. Files on a virtual disk drive can be backed up individually from i5/OS using file level backup with the Network Client (QNTC) file system in the IFS or using a native Windows backup application. See Chapter 7, “Backup and Recovery” on page 255 for more information.
2. Even though storage spaces are allocated out of IFS, storage operations are not performed by IFS while the integrated server is varied on. This means that operations such as journaling are not enabled.

The network server storage space (NWSSTG) objects are basically the same as described in Figure 2-11 on page 20, except for the following:

- ▶ When linking the virtual disk drive to the NWSD, it is necessary to identify which of the NWSD's storage paths to use for the SCSI data flows for that virtual disk drive.
- ▶ You can choose a specific storage path, the multi-path group or let the default storage path be used.

2.6.2 IBM Director

i5/OS locates and manages hosted systems by sending commands to the service processor of the remote system over an Ethernet network via a LAN adapter installed in the i5/OS hosting partition. IBM Director is used for these functions and must be installed and running in the partition that is connected to the iSCSI network.

2.7 iSCSI security model

iSCSI technology leverages the low cost and familiarity of Ethernet and IP networking. The flexibility of Ethernet and IP networking allows iSCSI attached systems to share hardware, extend the range, and increase bandwidth by adding hardware. However, this familiarity and flexibility lead to a requirement for appropriate network security. Each of the different types of networks used by iSCSI attached systems has its own security considerations.

Important: Because IPsec encryption has not yet been implemented for System i5 iSCSI, we recommend that you set up the iSCSI network in a physically secure environment. In this book, we assume that the System i5, iSCSI switched network, and hosted systems are all physically secured.

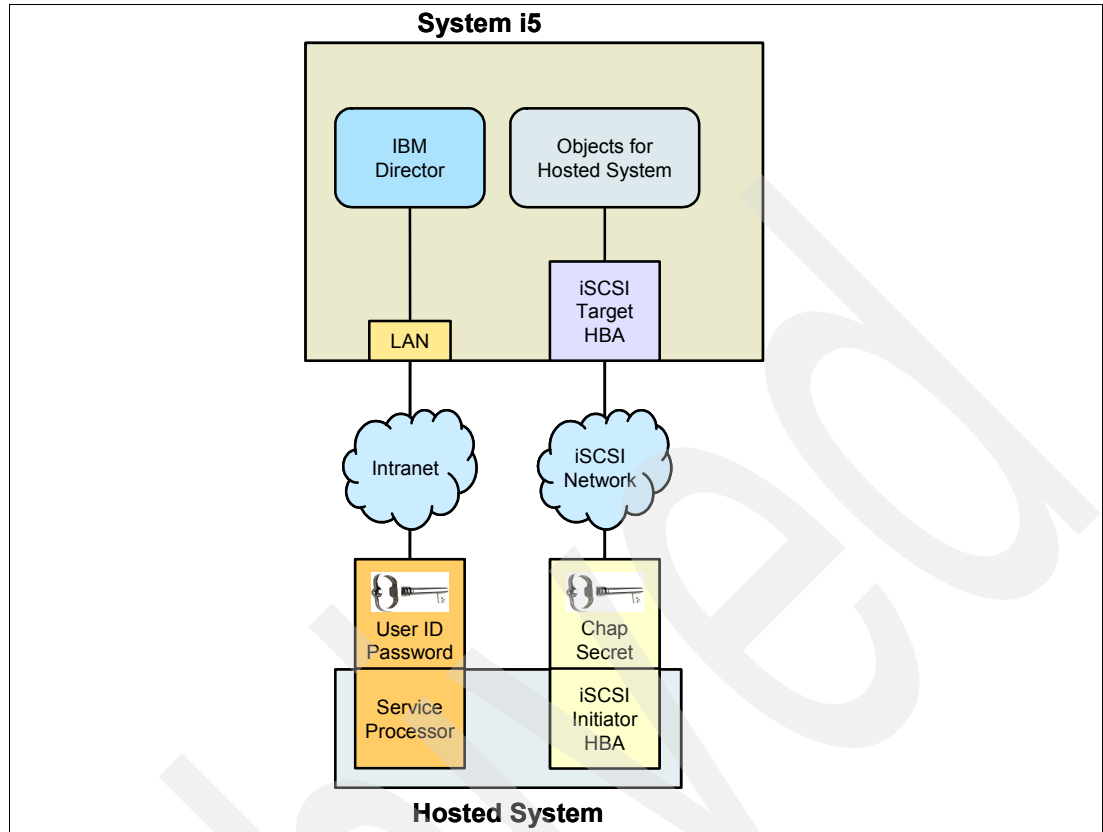


Figure 2-17 V5R4 iSCSI security model

2.7.1 Service processor connection security

Service processor security can involve one or more of the following mechanisms:

- ▶ Service processor password
- ▶ Network isolation and physical security

2.7.2 iSCSI network security

There are two types of iSCSI network traffic to consider:

- ▶ Storage security can involve one or more of the following mechanisms:
 - Challenge Handshake Authentication Protocol (CHAP)
 - Network isolation and physical security
- ▶ Virtual Ethernet security can involve one or more of the following mechanisms:
 - Network isolation and physical security

2.7.3 Service processor password

This password is managed by i5/OS and is used when your iSeries server starts a conversation with the hosted system's service processor. The service processor checks the password to ensure that the i5/OS configuration is authentic. New service processors have a default name and password. i5/OS provides a way to change the password.

2.7.4 Challenge Handshake Authentication Protocol (CHAP)

CHAP protects against the possibility of an unauthorized system using an authorized system's iSCSI name to access storage. CHAP does not encrypt network traffic, but rather it limits which system can access an i5/OS storage path. CHAP involves configuring a secret that both i5/OS and the hosted system must know. Short CHAP secrets can be exposed if the CHAP packet exchange is recorded with a LAN sniffer and analyzed offline. The CHAP secret should be random and long enough to make this method of attack impractical. i5/OS can generate an appropriate secret. A hosted system uses the same CHAP secret to access all of its configured i5/OS storage paths. CHAP is not enabled by default, but we strongly recommend you enable it.

2.7.5 Network isolation and physical security

Network isolation minimizes the risk of data being accessed by unauthorized devices and data being modified as it traverses the network. You can create an isolated network by using a dedicated Ethernet switch or a dedicated virtual local area network (VLAN) on a physical VLAN switch/network. When configuring a VLAN switch, treat an iSCSI HBA that is installed in your iSeries server as a VLAN-unaware device. Physical security involves physical barriers that limit access to the network equipment and the network endpoints at some level (locked rack enclosures, locked rooms, locked buildings, and so on).

2.8 IP addressing structure

A hosted server can have multiple physical iSCSI HBA ports. An iSCSI HBA port can carry traffic for iSeries storage paths, Virtual Ethernet networks, or both. A number of factors influence the nature of the traffic that flows through each iSCSI HBA port on the Windows server. IP addresses for iSCSI HBA ports can have a SCSI IP address, a LAN IP address, or both. A port with a SCSI IP address is a candidate for carrying storage traffic. A port with a LAN IP address is a candidate for carrying Virtual Ethernet traffic.

Note: Each iSCSI HBA interface can have two IP addresses, one for storage and one for LAN function, which is used to tunnel Virtual Ethernet. i5/OS TCP/IP is not aware of these IP addresses. For iSCSI HBAs, Virtual Ethernet is tunneled through a physical network with iSCSI HBAs at the physical endpoints.

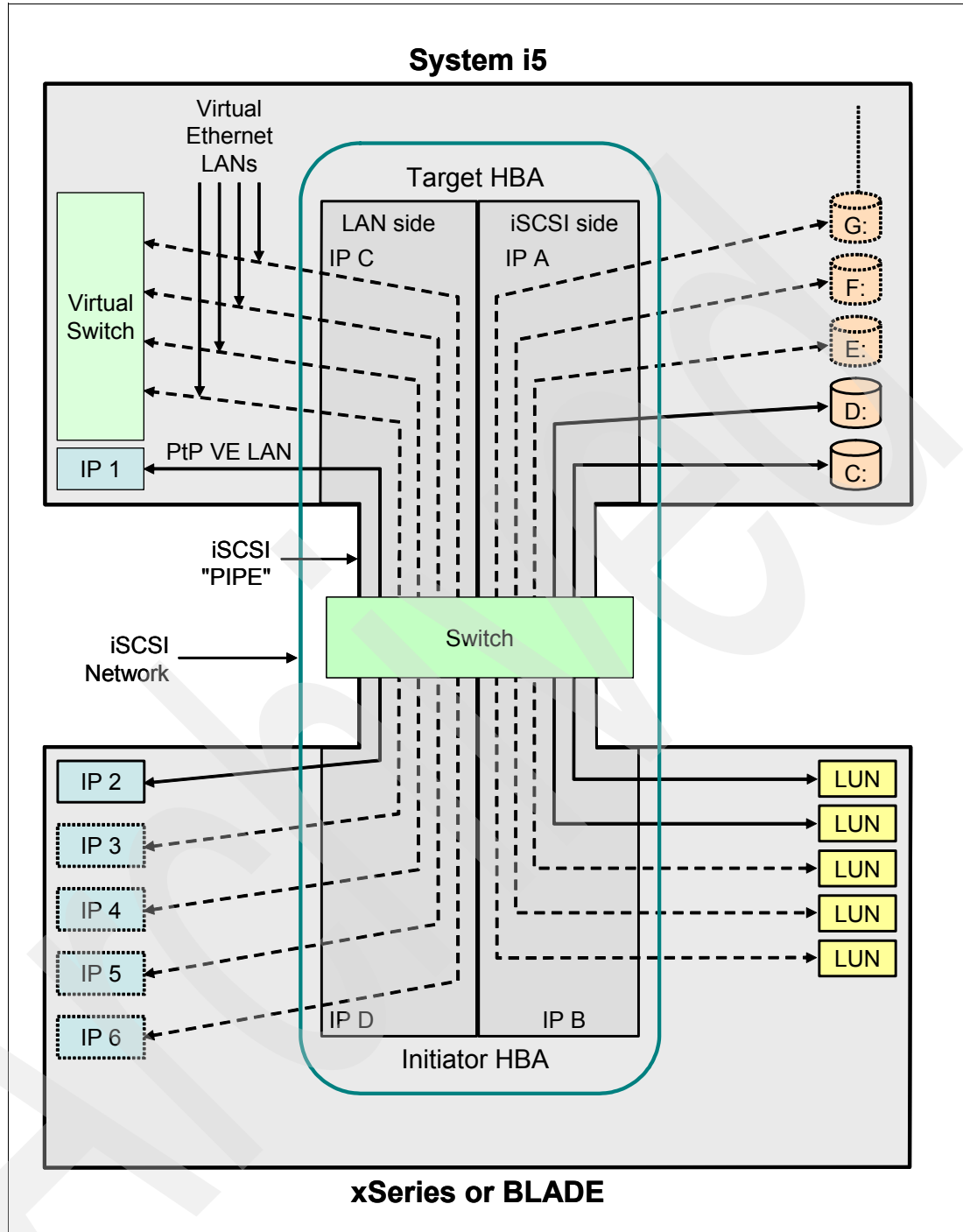


Figure 2-18 iSCSI network IP addressing structure

This physical network connects Ethernet iSCSI adapters in the hosting i5/OS with Ethernet iSCSI adapters in the hosted system. It is typically a simple, switched, Gigabit Ethernet network. Two kinds of traffic flow over this connection: storage (SCSI) and Virtual Ethernet (LAN). On one side of the network is an iSCSI adapter or adapters controlled by i5/OS. Each iSCSI adapter has two IP addresses: one for SCSI and one for LAN. You configure the IP addresses and other attributes of an adapter in an i5/OS device description object known as the network server host adapter. Each iSCSI adapter controlled by i5/OS needs its own object.

Every iSCSI adapter contains a TCP/IP stack implemented in hardware that is independent of the normal i5/OS TCP/IP stack. When you vary on a network server host adapter, an iSCSI adapter controlled by i5/OS uses the configured values. If you want different values to take effect, you must change the configuration and vary on the server host adapter again. The i5/OS TCP/IP stack is unaware of the IP addresses configured for the iSCSI adapters. On the other side of the network is an iSCSI adapter or adapters for the hosted system. You configure the IP addresses and other attributes of these adapters in an i5/OS object known as the remote system configuration. For more information, see Figure 2-11 on page 20. This configuration differs from the i5/OS network server adapter object in several ways:

- ▶ You can configure an iSCSI adapter port in a hosted system with 1 or 2 IP addresses: SCSI, LAN, or both. There must be at least one SCSI and one LAN IP address among all of the configured adapters.
- ▶ Whenever you configure an IP address for an iSCSI adapter in a hosted system, you must also configure the corresponding adapter MAC address. Each adapter has a label that shows its MAC addresses. Be careful to configure MAC addresses correctly.
- ▶ You configure all of the iSCSI adapters for a hosted system in the same i5/OS remote system configuration object. When the integrated server is subsequently varied on, the product automatically ensures that iSCSI adapters in the hosted system are using values in the i5/OS remote system configuration. If you want different values to take effect, you must change the configuration and vary on the server again.
- ▶ SCSI traffic uses the iSCSI adapter's hardware TCP/IP stack, but LAN traffic uses the Windows TCP/IP stack. Consequently, the Windows TCP/IP stack is unaware of the SCSI IP address, but is aware of the LAN IP address.

2.9 Hot spare

Server integration and storage virtualization provide innovative options that can enhance the reliability and recoverability of the Windows server environment. Hot spare hardware provides a way to quickly recover from certain types of hardware failures. This can reduce the server downtime from hours or days to minutes. With hosted systems, there are two ways to use hot spare hardware to minimize downtime that is caused by hardware failures:

- ▶ Hosted system hardware, including Integrated xSeries Servers, xSeries servers that are attached via an Integrated xSeries Adapter, and xSeries or IBM BladeCenter servers that are attached via an iSCSI host bus adapter, can be hot-spared. If the hardware that is used to run the hosted system fails, you can quickly switch the hosted system's disk images to compatible spare hardware and restart the hosted system.

If the Windows server hardware fails, you can quickly and easily switch the server's configuration to another hot spare xSeries server or IBM BladeCenter server without restarting your iSeries server. This can reduce the overall number of PC servers needed to provide increased availability. Hot spare support also adds flexibility by enabling one spare server to be used to protect multiple production servers.

Because the xSeries or IBM BladeCenter server that an iSCSI attached server runs on is simply defined via the remote system configuration name in the NWSD, it is easy to switch the hardware that an iSCSI attached integrated server runs on. By changing the remote system configuration name, the xSeries or IBM BladeCenter server that the existing NWSD is booted on can be hot-spared.

- ▶ For iSCSI attached servers, the iSeries target iSCSI host bus adapters (iSCSI HBA) can be hot-spared. If an iSCSI HBA that a hosted system is using fails, you can quickly switch the hosted system to use a spare iSCSI HBA and restart the hosted system.

The integrated DHCP server is a key and integral component when implementing hot spares. The DHCP boot mode enables automatic deployment of the required parameters defined in the System i5 software objects, eliminating the need to manually configure a server when boot parameters (IP addresses and IQNs) change.

iSeries and xSeries integration and storage virtualization provide options that can enable you to enhance the reliability and recoverability of your Windows server environment. If a Windows server fails, you can quickly and easily switch the server's storage spaces to another hot spare xSeries server without restarting your iSeries server. This might reduce the total number of Intel servers needed to provide increased availability. It also adds flexibility by enabling one spare server to be used to protect multiple production servers. The procedures for hot sparing an integrated server's hardware are shown below.

Using iSeries Navigator

These are the steps to follow:

1. Expand **Integrated Server Administration**.
2. Select **Servers**.
3. If the server for which you want to swap hardware is not already shut down: Right-click the server and select **Shut Down**. Click **Shut Down** on the confirmation panel.
4. Change the server configuration to point to the hot spare server hardware:
 - a. Right-click the server and select **Properties**.
 - b. Select the **System** tab and change one of the following:
 - i. For non-iSCSI servers, select the new Resource name and type.
 - ii. For iSCSI servers, select the new Remote system configuration name.
 - iii. Click **OK**.
5. To start the integrated server, right-click the server and select **Start**.

Using the green screen interface

These are the steps to use a green screen interface:

1. If the server for which you want to swap hardware is not already varied off, use the Vary Configuration (VRYCFG) command to vary it off.
2. To change the server configuration to point to the hot spare server hardware, use the Change Network Server Description (CHGNWSD) command to change one of the following:
 - a. For non-iSCSI servers, change the value for the Resource name (RSRCNAME) parameter to specify the new IXS or IXA hardware resource name.
 - b. For iSCSI servers, change the value for the Remote system name element of the Network server configuration (NWSCFG) parameter to specify the new remote system network server configuration object name.
3. To start the integrated server, use the Vary Configuration (VRYCFG) command.

2.9.1 Integrated DHCP server

The iSCSI attached server uses an integrated DHCP server when it is configured to use the default or DHCP boot mode. This integrated DHCP server is not a general purpose server. It is intended to exclusively deploy boot parameters to the hosted server iSCSI HBA. The server is automatically configured with the parameters provided in the remote system configuration when a network server description (NWSD) is varied on.

DHCP and DHCP relay

There are several methods for delivering boot information to the hosted system. The default method of delivering IP and storage information to boot Windows uses an integrated Dynamic Host Configuration Protocol (DHCP) server on i5/OS side of the iSCSI network. Even with DHCP, the IP address might be considered static because the DHCP server associates a single IP address with a MAC address. The integrated DHCP server is designed to coexist with any DHCP servers that might also be on the iSCSI network. If the iSCSI network includes routers between the iSeries server and the hosted system, and the boot information delivery method is DHCP, then an appropriately configured DHCP relay agent, also known as a *BOOTP relay agent*, is required in the network.

The iSeries iSCSI attached server solution provides an integrated DHCP server. The server is used to deploy boot parameters to the hosted system iSCSI HBA when the Dynamically delivered to the remote system via DHCP option is specified in the i5/OS remote system configuration object and AUTO or DHCP mode is specified in the hosted server iSCSI HBA. The integrated DHCP server is not a general purpose server. It is intended to exclusively deploy boot parameters to the hosted server iSCSI HBA. The server is automatically configured with the parameters provided in the remote system configuration when a network server description (NWSN) is varied on. The DHCP server only responds to the hosted server's iSCSI HBA DHCP client. All of the iSCSI HBA DHCP client requests use an IBM defined vendor id. The server is programmed to respond to requests that use the default vendor id. Any other requests from other devices in the network are ignored by the DHCP server. Providing the MAC addresses of the hosted server iSCSI HBAs in the remote system configuration object is very important. In addition to the vendor id previously described, the integrated DHCP server uses the MAC address to properly deploy boot parameters. MAC address is part of the specific scope required to ensure proper parameter deployment. The scope provided by the vendor id and MAC address can be changed. While this is considered an advanced function, provisions have been put in place to allow the advanced and sophisticated users to more specifically configure this setting, when required. The default vendor id can be configured to other values. Configuration screens are available in the hosted server iSCSI HBA adapter CTRL-Q setup utility and the corresponding remote system configuration object. This advanced function is compliant with the RFC 2132 specification. For more details about advanced configurations, see iSCSI install readme first.

When an incoming DHCP request is received by the integrated DHCP server and all of the required scope is matched, the integrated DHCP server provides to the DHCP client the IP addresses for the boot target device. The boot target device is the network server host adapter (NWSH) where the boot virtual disk is configured. The server also provides the IP address for the initiator or DHCP client. The initiator is the iSCSI HBA in the hosted server that will be used to boot over iSCSI. In addition, the integrated DHCP server provides the globally unique iSCSI Qualified Names (IQNs) that represent the target and initiator devices to the hosted system iSCSI HBA.

Both of these sets of IP addresses and IQNs are in the iSeries configuration objects used to define the hosted server. The target IP address is defined in the NWSH object. The initiator IP address and initiator IQN are defined in the remote system configuration object. The target IQN is automatically configured and defined in the NWSN object. For more information about these objects, refer to Figure 2-11 on page 20. The integrated DHCP server is a key and integral component when implementing hot spares. The DHCP boot mode enables automatic deployment of the required parameters defined in the iSeries software objects, eliminating the need to manually configure a server when boot parameters (IP addresses and IQNs) change.

2.10 General performance considerations

The following performance considerations apply to all the scenarios described in this chapter.

The iSCSI network is a 1Gb Ethernet switched network. However, the throughput depends on many factors including:

- ▶ The capacity and latency of the switches on the network.
- ▶ The number of target HBAs in the i5/OS partition.
- ▶ The number of initiator HBAs in the PC servers.
- ▶ The disk activity of the Windows applications running on the integrated servers.
- ▶ The Virtual Ethernet LAN activity of the Windows applications running on the integrated servers.
- ▶ The ratio of target HBAs to initiator HBAs for each xSeries and Blade server connection.
- ▶ A Blade server initiator HBA has two ports. These ports must be connected to different switch modules on the BladeCenter. The second port can provide additional flexibility and throughput on the initiator HBA. We show how you can use the second port in some of the scenarios below.

2.10.1 System i5 storage spaces

The following performance considerations apply to all the scenarios described in this chapter.

The iSCSI network is a 1Gb Ethernet switched network. However, the throughput depends on many factors including:

- ▶ The capacity and latency of the switches on the network.
- ▶ The number of target HBAs in the i5/OS partition.
- ▶ The number of initiator HBAs in the PC servers.
- ▶ The disk activity of the Windows applications running on the integrated servers.
- ▶ The Virtual Ethernet LAN activity of the Windows applications running on the integrated servers.
- ▶ The ratio of target HBAs to initiator HBAs for each xSeries and Blade server connection.
- ▶ A Blade server initiator HBA has two ports. These ports must be connected to different switch modules on the BladeCenter. The second port can provide additional flexibility and throughput on the initiator HBA. We show how you can use the second port in some of the scenarios below.

2.10.2 iSeries storage spaces versus dedicated disks

For performing processor or memory intensive work on an integrated server, the performance characteristics are equivalent to a stand-alone server using dedicated disk drives. Because the integrated server disk drives are allocated out of iSeries storage, the disk performance is dependent on the iSeries.

Consider the entire group of disks when you evaluate storage bottlenecks. The iSeries server storage space appears as one disk drive within Windows. When the Physical Disk average queue length (in Windows Performance Monitor) exceeds two, the server performance is not necessarily disk-constrained. Assuming that memory paging issues have been ruled out, a queue length of two or a Windows disk utilization of 100% only points to a storage bottleneck if there is only one physical disk drive to perform the operations. There are usually multiple disks on the iSeries server in the storage space ASP operating in parallel. Typically, two times the number of disks in the ASP might point toward a disk bottleneck. You might also need to account for the average queue lengths of all the servers using the storage ASP.

2.10.3 iSCSI server performance

For iSCSI attached servers, there are multiple configuration options to adjust for better performance capacity as needed. Some options might require different target disk configurations or volumes on the integrated servers.

Using Windows disk configuration for iSCSI attached integrated servers, the virtual disk drives are optimized for:

- ▶ One disk partition per virtual drive.
- ▶ One gigabyte or larger storage spaces.
- ▶ NTFS file system formatted with four kilobyte or larger cluster sizes.

These guidelines allow the iSeries to efficiently manage the storage space memory, improving the disk performance. These guidelines also affect IXS attached and IXA attached servers, but to a significantly smaller degree. If you use the Change Network Storage Space (CHGNWSSTG) CL command to increase a storage space size, be sure to use the Windows Server 2003 DISKPART command to also increase the size of the partition on Windows.

For better performance, add a storage space to the server instead of adding another disk partition in the new space. If using iSeries memory pools for iSCSI attached servers, the storage operations occur through an iSeries memory pool. This memory essentially acts as a cache to the disk operations, so the size of the memory can affect the Windows disk performance. This I/O does not directly cause page faulting in the base pool. However, because pool memory is shared with other i5/OS applications, Windows disk operations can cause page faulting in other applications, or other applications can induce paging of iSCSI disk operations. In extreme cases, you might need to adjust memory pool sizes or assign applications to other memory pools to mitigate memory problems. IXS attached and IXA attached servers do not perform disk operations through a base memory pool. They use reserved memory within the machine pool (System Pool ID 1). Thus, the disk operations do not share memory with other applications. For iSCSI performance configurations on iSCSI attached integrated servers, if a single network fabric is reaching capacity, you can add channels with additional iSCSI HBAs in both the xSeries and iSeries servers (assuming the interconnecting network also has available bandwidth).

There are several ways that you can spread the iSCSI and network traffic between the separate channels:

- ▶ Dedicate SCSI operations to one channel and Virtual Ethernet operations to another.
- ▶ Use two storage targets. Each target should be linked to a separate HBA path:
 - On Windows, direct applications to use both drives (if possible), or dedicate the drives to different applications to spread the total disk operations between the drives.
 - Configure the two disks in a Windows dynamic volume set with the data striped across the two drives. As applications use the volume, the disk operations automatically balance across the drives in the volume set.

2.10.4 Managing hosted system iSCSI adapter utilization

You can configure an iSCSI adapter in a hosted system with a SCSI IP address, a LAN IP address, or both kinds of IP addresses. The presence of a SCSI IP address enables storage traffic, and the presence of a LAN IP address enables Virtual Ethernet traffic. Each Windows Virtual Ethernet adapter is normally automatically assigned to a physical iSCSI adapter. There is an option on the advanced properties tab of each Virtual Ethernet adapter that allows a particular physical iSCSI adapter to be selected.

Note: IBM does not support the use of the iSCSI adapter as a general purpose external network connection.

2.10.5 High bandwidth and low latency is desirable for the iSCSI network

Virtual Ethernet can take advantage of an MTU up to a 9000 byte “jumbo” frame if the network supports the larger MTU. This improves Virtual Ethernet performance.

2.10.6 Maximum transmission unit (MTU) considerations

The frame sizes discussed here do not include the Ethernet 14 byte MAC header. In contrast to the 9000 byte jumbo frames provided in IXS attached and IXA attached servers, Virtual Ethernet in systems attached by an iSCSI network defaults to a smaller frame size that can be transported in a standard 1500 byte Ethernet frame. If the iSCSI network is capable of a larger frame size, you can configure Virtual Ethernet to use a larger size up to 9000 bytes, which improves performance. In a complex iSCSI network, there can be a mix of maximum frame sizes depending on the network topology and equipment involved.

Configuring Virtual Ethernet

This section discusses configuring Virtual Ethernet for maximum performance on iSCSI networks that support frames larger than 1500 bytes and less than 1500 bytes.

iSCSI networks that support frames larger than 1500 bytes

This section discusses configuring Virtual Ethernet for maximum performance on iSCSI networks that support frames larger than 1500 bytes.

At the Windows console, perform the following steps:

1. Navigate to the Network Connections Window.
2. Double-click the iSCSI adapter that is connected to the iSCSI network supporting frames larger than 1500 bytes.
3. Click **Properties**.
4. Click **Configure**.
5. Click **Advanced**.
6. Click **Ethernet Frame Size**.
7. Select a value that is as large as possible without exceeding the iSCSI network's maximum frame size.

Note: Related configuration items listed below should be left at their default values:

- ▶ For Windows Virtual Ethernet adapters, Maximum Frame Size defaults to Auto. Auto causes Virtual Ethernet to calculate a maximum frame size based on the Ethernet Frame Size of the iSCSI HBA port used.
- ▶ In i5/OS Virtual Ethernet line descriptions, Maximum frame size (MAXFRAME) defaults to 8996. In i5/OS TCP/IP interfaces for Virtual Ethernet, Maximum transmission unit (MTU) defaults to *LIND.

iSCSI networks that have a maximum frame size less than 1500 bytes

This section discusses configuring Virtual Ethernet for iSCSI networks that have a maximum frame size that is less than 1500 bytes.

At the Windows console, perform the following steps:

1. Navigate to the Network Connections Window.
2. Double-click the IBM iSeries Virtual Ethernet x adapter that will use a iSCSI HBA connected to the iSCSI network having a maximum frame size less than 1500 bytes.
3. Click **Properties**.
4. Click **Configure**.
5. Click **Advanced**.
6. Click **Maximum Frame Size**.
7. Select a value that is as large as possible without exceeding the iSCSI network's maximum frame size.

Configuring Virtual Ethernet for unusual non-TCP applications

This section discusses configuring Virtual Ethernet to support unusual non-TCP applications that do not negotiate MTU.

Note: To avoid impacts to normal applications that negotiate MTU, before performing this procedure, you might want to define a separate Virtual Ethernet network or separate IP addresses for the application that does not negotiate MTU.

1. Perform one of the following:
 - a. If all Windows endpoints will use an iSCSI network having a maximum frame size of 1500 bytes or greater, configure the iSCSI HBA Ethernet frame size at all Windows endpoints to a value as large as possible without exceeding the most constrained iSCSI network's maximum frame size.
 - b. If any Windows endpoint will use an iSCSI network having a maximum frame size less than 1500 bytes, configure the Virtual Ethernet Maximum frame size at all Windows endpoints to a value as large as possible without exceeding the most constrained iSCSI network's maximum frame size.
2. At other endpoints, set the MTU to a value determined by subtracting 116 from the smaller of the Windows iSCSI HBA Ethernet frame size and the Virtual Ethernet Maximum frame size. For i5/OS endpoints, you can accomplish this by performing the following procedure:
 - a. Using iSeries Navigator, expand **Network** → **TCP/IP Configuration** → **IPv4** → **Interfaces**.
 - b. Right-click the interface with the IP address and line description name of interest and select **Properties**.

- c. On the Advanced tab, type the calculated value in the Maximum transmission unit field and click **OK** to save the change.

Note: If you want to use the command line interface, use CFGTCP and select option 1, Work with TCP/IP interfaces.

Archived

Archived

Planning for iSCSI attached servers

The iSCSI integrated server is a more complex product to install and configure than its predecessors. It provides more capability and flexibility than the earlier implementations, but it introduces new elements that will require you to plan carefully before setting up the environment. There are new concepts, terminology, configurations, and connections for this product that have no analogues in the other implementations and therefore require that you understand these concepts and issues before proceeding. Also, if you take the time to read the available materials, you might be able to expand the flexibility and scalability of your environment and utilize hot spare options to protect your data.

This chapter discusses the various considerations involved in preparing to install and configure your servers. It discusses:

- ▶ Resources for planning
- ▶ Pre-planning: Learn about the product
- ▶ Planning for System i5
- ▶ Planning for initiator systems
- ▶ Network planning
- ▶ Security

3.1 Understanding the environment

If you are new to the integrated server environment, you should review the information about the Web site for the integrated servers first at:

<http://www.ibm.com/servers/eserver/series/integratedxseries/windows/>

Also, read *Windows environment on iSeries* (part of iSeries infocenter) for an understanding of the fundamentals of the integrated server and then proceed to the specific documentation for the iSCSI integrated server:

<http://publib.boulder.ibm.com/infocenter/series/v5r4/topic/rzahq/rzahq.pdf>

If you have prior experience with the IXS/IXA servers you **MUST** still read the iSCSI Install readme prior to attempting an install because this environment cannot be installed with only a knowledge of the prior products! There are a number of differences with this product that you must be aware of in order to successfully install and configure an iSCSI network and iSCSI server. See also the discussions in Chapter 1, "Introduction to iSCSI integrated server support" on page 1, as well as the readme for information about new hardware connectivity and new objects to describe the hardware.

Understanding the environment will be necessary to order the components that will serve you best or to insure that you have all of the pieces and that they are supported with the integrated server. Furthermore, the various components cannot all be ordered from one source so you will want to plan accordingly. See below for information ordering and supported hardware.

Whether you require one or more xSeries server or whether you choose to order a BladeCenter and one or more blades depends on your particular needs and can be influenced by the capabilities of the server hardware. We do not present a detailed analysis of each of the initiator products, but we point out several considerations within the context of the supported hardware.

3.1.1 Ordering the hardware

Figure 3-1 on page 45 is a list of hardware components and where to order them.

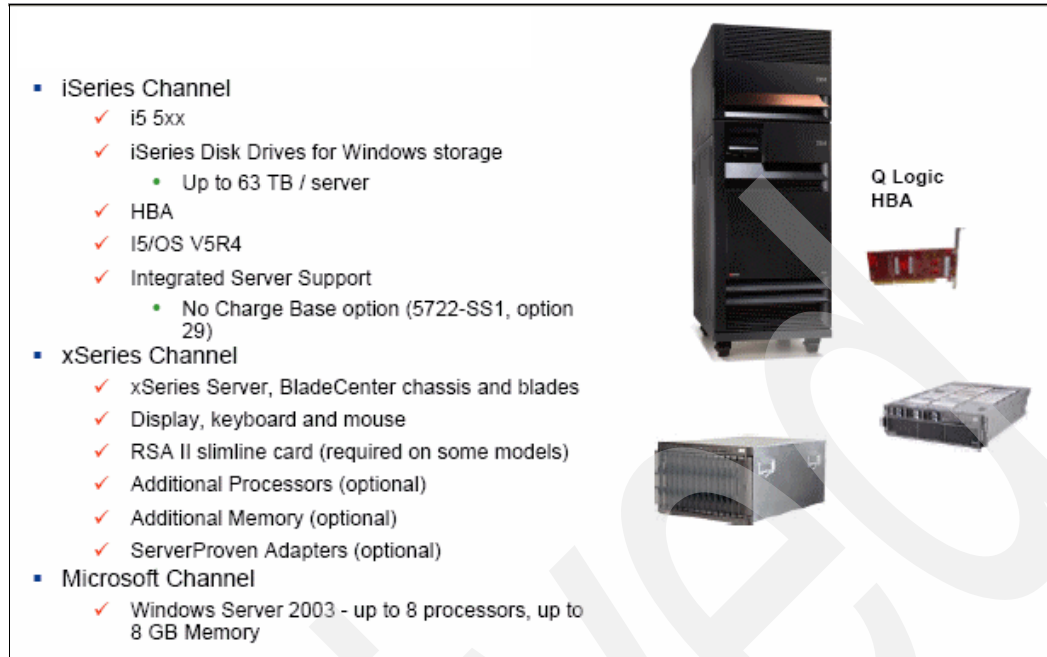


Figure 3-1 Channels for ordering iSCSI server components

It is not possible to document what hardware is appropriate for every environment, but here are some notes which might help you decide. These are general notes which we follow with planning sections for specific issues related to the i5, xSeries, and Blade Center:

- ▶ As with all integrated servers, the server hardware chosen must not have any hard drives. Any disk drives or disk drive controllers in the server hardware would need to be removed.
- ▶ Many xSeries servers are supported. Most of them require an RSA adapter. There are several models of these, so verify you have a supported model. xSeries servers with more than eight processors are not supported.
- ▶ With regard to BladeCenters, there is a “break even” point where they become cost-effective so if you only need one blade, this might not be the solution for you.
- ▶ Currently supported blades only have a maximum of two processors so if you need a larger number of processors, you need to order an xSeries server.
- ▶ If your server will need multiple ports, be aware that the daughter card for a Blade provides two ports whereas an xSeries server can have up to four. The four HBAs might not be practical depending on how many expansion slots are being utilized. Also, the second port on the BladeCenter requires an additional switch to enable it.

3.1.2 Planning for System i5 hardware

Planning for System i5 hardware is relatively straightforward if you are familiar with the system. The following considerations apply:

1. The iSeries must be a System i5 models 520, 550, 570, and 595. This support will not be delivered on older models.
2. Although not directly related to iSCSI integration, if the system is going to be partitioned, you want to consider where your server or servers are located because there are specific software requirements for each partition that hosts integrated servers. Partitioning can also affect your networking scheme (you might need to plan for inter-LPAR Virtual Ethernet as well as other communications between partitions and it might also affect the

resources that you allocate to the partition as well. The location of switches and network hardware is also affected by the relationship of iSCSI devices relative to partitions.

3. These servers can coexist with IXS and IXA server environments.
4. There are two supported adapter part numbers. Which of the two you select is dependent on what your network is like or what it will be like. If this will be a new network, the major criteria for choosing between the two is that fiber might be preferred in environments with a lot of electrical noise, or where longer cable runs are required. Although the adapters have one physical port, they support up to eight iSCSI connections. The adapters are as follows:
 - a. A copper Ethernet HBA adapter, which is feature code 5783. It is a one port PCI-X adapter (1 Gbps).
 - b. A fiber Ethernet HBA adapter, which is feature code 5784. It also is a one port PCI-X adapter (1Gbps).
5. The card is a standard PCI-X adapter which can run in 32 bit or 64 bit 3.3 volt slots. See the publication, *PCI and PCI-X Placement Rules for IBM System i5, eserver i5, and iSeries servers with i5/OS V5R4 and V5R3*, REDP-4011-02, which details the PCI placement rules to determine the appropriate slots on your machine. It recommends the card is put in 64 bit slots.
<http://www.redbooks.ibm.com/redpapers/pdfs/redp4011.pdf>
6. The cabling and switches.
7. In some environments, a consideration might be the maximum number of adapters that can be installed in your system. See Figure 3-2.

i5 Model	Maximum Host Bus Adapters (HBA)
520	21
550	42
570	84
595	168

Figure 3-2 Maximum HBAs for i5

3.1.3 Planning for i5 Software and Preparation

The System i5 has the following software requirements to support iSCSI integrated servers:

1. V5R4 of the i5 operating system.

Important: Integrated Server Support and IBM Director Server in particular and by extension the other programs listed need to be installed in any partition that is hosting iSCSI servers. Multiple Director Servers might have ramifications. See Chapter 5, “Implementing IBM Director Server” on page 107.

2. License Program Products:

- a. Extended Base Directory Support (5722-SS1 Option 03).
- b. Integrated Server Support (5722-SS1 Option 29).
- c. TCP/IP Connectivity Utilities (5722-TC1).
- d. Java™ Development Kit 1.4, Option 6 (5722-JV1).
- e. IBM Director 5.10, which is part of 5733-VE2. See Chapter 5, “Implementing IBM Director Server” on page 107.
- f. IBM i5/OS Digital Certificate Manager (5722-SS1 Option 34).
- g. IBM HTTP Server for iSeries (5722-DG1).
- h. IBM iSeries Access for Windows (5722-XE1) because there are many tasks that are much easier to do with iSeries Navigator.

Important: If you are using an older version of iSeries Navigator, you will need to upgrade to V5R4 version because earlier versions did not have the needed capabilities. The Integrates Server Administration simplifies many of the configuration tasks.

- i. QSHELL (5722-SS1 Option 31).

iSeries tasks to prepare for the install

Other tasks that are either necessary or desirable to do before installing include:

1. Load required PTFs.
2. Install V5R4 iSeries Navigator on a PC with at least the following components:
 - a. Configuration and Service
 - b. Network
 - c. Integrated Server Administration
3. Set the system value for Qretsvrsec to “1”. Although it has always been recommended that this system value is set to 1, it was not stated as required. iSCSI requires this value to save security information.
4. Be aware that the virtual disk, which formerly utilized the machine pool for memory, will now default to use QBASE. You might need to plan for this. Heavy I/O on the virtual disk might flush the cache. There is work being done to provide a PTF that will allow you to customize the environment.

3.1.4 Planning for xSeries

There are several xSeries server models available, which have different characteristics. The IXA implementation had a dedicated connection to the service processor, which the iSeries

accessed to power off and power on the server. The only recommended access to it was using a dedicated Personal Computer to flash its firmware. ASCII is different. There are two basic versions of service processor: BMC and RSA (which has several different types). Most of the models are required to have RSA adapters and these are noted in the supported models documentation.

xSeries servers can have up to four HBAs. This will, of course, depend on the number of available expansion slots. There might be a reason to have more than one HBA if you have certain bandwidth requirements. See Chapter 5, “Implementing IBM Director Server” on page 107, which is the scenarios chapter, for ideas about how multiple HBAs might be used to improve your bandwidth for storage or virtual LAN. Or perhaps you would have multiple adapters for hot spares. These adapters are copper or fiber such as the iSeries adapters.

The general requirements for an xSeries server are shown in Figure 3-3.

- **Hardware:**
 - ✓ Supported xSeries systems:
 - x236, x306 (TBD), x336, x346, x366 and x460: initial models
 - Open bay (diskless server): disk drives in the server are not supported due to possible boot disk conflicts.
 - ✓ Qlogic iSCSI HBA(s)
 - 30R5201 – copper
 - 30R5501 - fiber
 - 32/64 bit 3.3v PCI-X slot
 - Supports 1 to 4 HBA(s) per server
 - ✓ RSAAI Slimline Adapter 2 required on some models (see specific model details): some models will use their built in Baseboard Management controller (BMC) in place of the RSA II.
 - **Software:**
 - ✓ Windows Server 2003 with SP1: the service pack is a requirement and needs to be included on the CD media.
 - ✓ Intel Linux support: IBM has issued a statement of intent to add this support at a future date. Check our web site for updates.

Figure 3-3 xSeries requirements

It is mandatory to verify the firmware levels for the Xseries components and update them as necessary. These include:

- ▶ BIOS for the xSeries
- ▶ Drivers: Normally, you need to get the Broadcom drivers. Depending on the options that you purchased with the machine, you might need other drivers. The Integrated Server Support does not provide xSeries hardware drivers.

Note: BIOS and some of the drivers can be obtained via a single CD burned from the .iso image provided as Update Express from:

<http://www.ibm.com/products/finder/us/en/finders?pg=ddfindex>

- ▶ BMC firmware
- ▶ RSA firmware
- ▶ HBA - QLogic BIOS and firmware

You should always follow the readme or instructions for each of the updates, but for more information, you can also see the iSCSI Host Bus Adapter publication.

3.1.5 Planning for BladeCenter

BladeCenters have a large number of optional features that require research to determine what and how much of each that you need. They have options available for setting up redundant hardware and hot plug features.

The initial chore is which chassis to purchase. The chassis is the enclosure into which the blades and other modular pieces are installed. A chassis comes as a 7U form factor that substantially reduces the area that servers occupy. A chassis can have up to 14 blades installed. A 47U rack can accommodate six chassis making 84 servers available in a small area.

There are two power domains in a chassis. The first supports the first six slots and is supported by the two redundant power supplies. If you plan to populate any of the remaining slots, this requires the second two power supplies, which are otherwise optional.

The iSCSI implementation requires a particular model of chassis and of blade. Blades actually come with Intel, AMD, or power processors, but only the Intel Blade is supported.

Will the initiator and target be connected through a dedicated switch? Will this be an existing switch? There are a number of requirements for this switch:

- ▶ One gigabit layer 2 switch for the HBA connection.
- ▶ Copper or fiber ports depending on the model of HBAs being used.
- ▶ Switch ports should be “access ports”, not “trunk ports”.
- ▶ Enough ports to accommodate all connections.

Switch Considerations:

- ▶ Jumbo frames (9000 byte) provide the best performance, but are not required. The switch needs to be configured for this.
- ▶ Switches that support IEEE 802.1q VLAN should treat the HBAs and service processor connections as VLAN unaware.
- ▶ We recommend you have the service processor connection on another network, which does not need to be 1 gigabit. This connection will need to be on the switch that the System i5 connection is on because IBM Director needs to discover the processor. If it is on a private network, it will require a dedicated connection to be able to administer and access the administrative utilities.
- ▶ IBM BladeCenter switches:
 - BladeCenter provides network access via I/O modules. These can be copper or fiber switches that are installed into the chassis or pass-through modules.
 - If you have switch modules, you do not need external switches to connect to the iSCSI HBAs unless there are other considerations. The HBA connects directly to these switches.
- ▶ Additional features that provide additional capability for managing the switch:
 - Port mirroring to accommodate a sniffer. We recommend this for diagnostic purposes, because iSeries communication traces cannot trace this traffic.
 - Statistics support.

Service processor considerations:

- ▶ The service processor connection for iSCSI is a standard Ethernet connection. It must be connected to the network, which the i5 Ethernet connection is on, so that IBM Director Server can “discover it” and submit the request to power on or power off the server.

- ▶ Addressing the service processor. There is a default address, which is the same for virtually all IBM processors, so it will be necessary to change this if you have multiple servers. We recommend that this is a static address although it is possible to have it get its address from a DHCP server on the network.
- ▶ The default user ID and password are the same for all service processors in the xSeries and BladeCenter. In an open network, it is obviously desirable to change this for security reasons, but it is also important to change it because of the nature of IBM Director, which takes control of any service processor it can log in to and establish a session with.

3.2 Resources for planning

The following are several resources that you will want to utilize to plan for a successful iSCSI integrated server install. You might not need all of them, but if you are going to only look at one, make sure it is the first link for the readme.

At the time we are writing this, some of the links and publications might not be available and there might be changes that occur after this document is released, so you want to check the iSCSI install readme first at:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/>

You should read through this document in detail, because it presents a detailed process for learning about the product and installing a single server instance using the simplest possible scenario (one target and one port on one initiator). If you understand this scenario and the associated concepts, implementing the more complex environments is really just a matter of applying the concepts to multiple instances.

We provide a lot of the same information as the readme link, because you might not always have access to the Web information at a particular location, but you should refer to this site for changes or additions that might have occurred. As is stated in the readme document, the information in the readme document supersedes that which is located elsewhere.

The following are also valuable resources for planning:

1. For general integrated server information including supported models, supported versions of microsoft OS and tested service packs, latest fixes for the integrated servers, and much more, go to:
<http://www.ibm.com/servers/eserver/series/integratedxseries>
2. Windows environment on iSeries (part of iSeries infocenter) at:
<http://publib.boulder.ibm.com/infocenter/series/v5r4/topic/rzahq/rzahq.pdf>
3. IBM Director Installation and Configuration Guide at:
http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/fqp0_bk_install_gde.pdf
4. iSCSI Network Planning Guide:
http://www.ibm.com/systems/i/bladecenter/pdf/iscsi_planning.pdf
5. System p5™ and i5 eServer p5 and i5 and OpenPower™ PCI adapters at:
<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphak/iphak.pdf>
6. System p5 and i5 eServer p5 and i5 iSCSI Host Bus Adapter:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/>

7. Ethernet switches for iSCSi:

<http://www-03.ibm.com/systems/i/bladecenter/iscsi/switches.html>

8. For xSeries and Blade Center, see *Configuration and Options Guide*, SCOD-3ZVQ5W, at:

ftp://frp.software.ibm.com/pc/pccbbs/pc_servers_pdf/cog.pdf

9. For other BladeCenter specific information, see:

<http://www.ibm.com/systems/bladecenter/>

10. For xSeries specific information, see:

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

11. The IXA server hardware required drivers and firmware updates for the xSeries specific hardware and this was sometimes confusing. This is still true for xSeries, but also applies to Blades and BladeCenter. Using this link you can find both xSeries and BladeCenter downloads and it should simplify locating the appropriate driver or firmware:

<http://www.ibm.com/pc/support/site.wss/DRVR-MATRIX.html#DLP>

Important: Before beginning to plan, note that at the time of writing, Linux is not available on the iSCSI integrated server. Also, while reading the iSeries documentation, you might see references to the following topics, however, they are not currently supported:

- ▶ Multi-path I/O on the iScsi network
- ▶ Microsoft Cluster Service on iSCSI servers
- ▶ Secure Sockets Layer (SSL) security for the processor connection
- ▶ IP Security (IPSec on the iSCSi network)
- ▶ Routers on the iSCSI network

Archived

Installing the iSCSI integrated server

The INSWNTSVR command is used to install Windows Server 2003 onto an xSeries server or a Blade in an IBM BladeCenter. It does not support installing multiple servers with a single command, nor using IBM Director or Remote Deployment Manager (RDM) to deploy a single image to multiple servers. The utility for cloning images that was provided on the integrated xSeries server Web site has not been updated to support this process at the time of this writing.

In this chapter, we discuss:

- ▶ Pre-installation requirements
- ▶ Install command
- ▶ Objects created by the install command
- ▶ New objects that you are required to create
- ▶ Problem determination process

4.1 Pre-installation requirements

We do not reiterate what has already been discussed elsewhere with regard to pre-installation tasks. The intention of this section is to remind you that these tasks should have been accomplished prior to running the install command and creating the associated objects. This section assumes that the hardware has been installed and cabled and that the operating system is at V5R4M0. You should have already completed or must complete the following tasks before you can successfully execute the install command:

- ▶ You must read thoroughly the *iSCSI Install readme first* and printed or saved the referenced documents.
- ▶ You must verify that the Integrated Server Support and IBM Director Server and its associated programs have been installed. The latest cumulative and group fixes should be installed as well as any pertinent Director fixes. If you installed the Integrated Server Support after the operating system and cumulative PTF tape, be sure to install the PTFs for the licensed program product (LPP) before proceeding.
- ▶ You should have completed the other i5 preparations documented in the planning section.

Note: Check the following Web site for any PTFs that might not be on a cumulative tape for the Integrated Server Support. At the time of this writing, there were no service packs available:

<http://www-03.ibm.com/servers/eserver/series/integratedxseries/sp.html>

- ▶ The firmware for your xSeries, BladeCenter, management module, switches, and blades should have been verified to be current and upgraded if necessary.
- ▶ You should have performed the configuration for the service processor, HBA, and hardware per the iSCSI Host Bus Adapter document. While doing this, you should also have noted the MAC addresses for the SCSI and Lan interfaces of the HBA and written them into the worksheet from the iSCSI Network Planning Guide:
http://www-03.ibm.com/systems/i/bladecenter/pdf/iscsi_planning.pdf
- ▶ Additional configuration that you can perform is to configure a CHAP name and CHAP secret for the initiator per the same document. If you do configure CHAP, note the CHAP name and secret, because you will need to specify this in the install command.
- ▶ Verify that the version of Windows you are using is Windows Server 2003, that it is a supported edition, and that the Windows Server 2003 media includes SP1. It is important the install media have the SP1 integrated with it.
- ▶ Create the Network Server Host Adapter Device and vary it on. See the discussion at “New configuration objects required for iSCSI” on page 59.
- ▶ Consider pre-creating the NWSCFG objects. See the discussion at “New configuration objects required for iSCSI” on page 59.

4.2 The install Windows server command

The install Windows server command (*INSWNTSVR*) has been used to install the integrated servers since the advent of the integrated server support for the Microsoft operating system. The command is one of the unique features of the integrated server. The install is launched from the command line of the System i5 and begins by running on the System i5 where it creates the NWSD and other objects on the i5 system including the system and install source

drives after which it kicks off an unattended install file and passes control to the Windows operating system to complete the install.

The command to install IXS/IXA servers is the same command for iSCSI servers, however, there are more requirements prior to doing the install for iSCSI and there is additional configuration required on the System i5.

Important: The install command formerly created all of the necessary i5 objects required, however, for iSCSI, there is one object (NWSH) that has to be created and varied on prior to running the install and there are other objects which we recommend you precreate, but which can be created using the install command. Additionally, IBM Director Server must be started prior to running the command.

4.2.1 Objects created by the install command on System i5

The *INSWNTSVR* command creates all of the same objects that it has in the past. However, it can optionally create new objects that are required to discover the xSeries or BladeCenter and additional Virtual Ethernet connections as well. The objects required to communicate with the server hardware are of the type **NWSCFG*. We do not recommend you create these objects with the INSWNTSVR command. See 4.2.4, “Reasons for precreating *nwscfg objects prior to the install” on page 77. The objects created by default are:

1. An *NWSD* (network server description), which describes an instance of a server. It links together various objects that describe the iSCSI server. Many of the elements will be familiar to you if you used the integrated server previously, but there are many new elements that are associated with the new methodology of locating and accessing storage and Virtual Ethernet. We discuss them later. The following diagram shows how all of the configuration contained in the NWSD is related. Following the diagram we have inserted examples of new elements contained within the NWSD relative to iSCSI, such as Figure 4-1 through Figure 4-5 on page 57.

Display Network Server Desc		02/26/06 12:51:58
Network server description	ANWSD	
Option	*BASIC	
Resource name	*NONE	
Network server type		
Server connection	*ISCSI	
Server operating system	*WIN32	
Activation timer	120	
Vary on wait	*NOWAIT	
Shutdown timeout	5	
Domain role	*SERVER	
Propagate domain users	*YES	
Language version	2924	
Code page	850	

Figure 4-1 DSPNWSD of type iSCSI

```
Display Network Server Desc
02/26/06 13:05:11
Network server description . . . . : ANWSD
Option . . . . . : *BASIC

Network server configuration . . . :
  Remote system name . . . . . : AAS300RS
  Library . . . . . : QUSRSYS
  Connection security name . . . . : AAS300CN
  Library . . . . . : QUSRSYS
Synchronize date and time . . . . : *YES
Shutdown TCP port . . . . . : 8700
Virtual Ethernet control port . . : 8800
```

Figure 4-2 DSPNWSD of type iSCSI displaying links to new configuration objects

```
Display Network Server Desc
02/26/06 12:59:03
Network server description . . . . : ANWSD
Option . . . . . : *VRTETHPTH
Virtual Ethernet path . . . . . :

-----Host adapter VE paths-----
Port      Host      Path
number    name      resource
*VRTETH3   AS300NWSH1 *NONE
*VRTETHPTH AS300NWSH1 *NONE
```

Figure 4-3 DSPNWSD of type iSCSI displaying new Virtual Ethernet Path

```
Display Network Server Desc
Network server description . . . . : ANWSD
Option . . . . . : *STGPPTH
Default storage path . . . . . : 1
Removable media path . . . . . : 1
Multi-path group . . . . . : *NONE

Storage path . . . . . :

-----Host adapter storage paths-----
Path      Host      Path      Path
number    name      resource  status
1          AS300NWSH1 *NONE     VARIED OFF
2          AS300NWSH2 *NONE     VARIED OFF
```

Figure 4-4 DSPNWSD of type iSCSI showing new storage path

Display Network Server Desc			02/26/06 12:59:03
Network server description	:	ANWSD	
Option	:	*STGPTHIQN	
Multi-path IQN	:		
Storage path	:		
-----Host adapter storage paths-----			
		iSCSI	
Path	Host	qualified	
number	name	name	
1	AS300NWSH1	iqn.1924-02.com.ibm:10a358b1.anwsd.t1	
2	AS300NWSH2	iqn.1924-02.com.ibm:10a358b1.anwsd.t2	

Figure 4-5 DSPNWSD of type iSCSI displaying iSCSI-qualified name

Important: Remember when choosing a name for the nwsd that the default is to install the Windows server with the same name. The server name can be changed in Windows, but the nwsd name cannot be changed and is limited to eight characters.

2. A *Virtual Ethernet (VE) point-to-point* line description is created which will have a name that is the NWSD name with a PP appended to it. See Figure 4-6. The line description looks the same as it did with IXS/IXA if you examine the line description itself. Within the NWSD, however, you see references to a host adapter VE path. This indicates which HBA the Virtual Ethernet connections are using as a channel. Figure 4-8 on page 58 shows one of the connections that is new with iSCSI. Previously, the Virtual Ethernet point-to-point had been implemented internally to the System i5 and did not utilize LAN hardware at all. With iSCSI, the Virtual Ethernet is a tunneled connection utilizing one or more of the host bus adapters. See 8.1, “Introduction to Virtual Ethernet LAN (VE LAN)” on page 306.
3. A *TCP/IP interface* for the point-to-point line, which communicates with the LAN interface on the Windows server. A “*Local Area Connection*” in Windows that is created by the install command. Previously, the interface defaulted to the form of 192.168.X.Y where X was the hardware resource of the IXS or IXA and Y was initially a value of 1 for the iSeries side and 2 for the Windows side of the connection. The defaults for iSCSI are 192.168.100.1 (iSeries) and 192.168.100.2 (Windows).

192.168.100.1	255.255.255.0	BNWSDPP	*ELAN
---------------	---------------	---------	-------

Figure 4-6 Example of default TCP/IP interface for iSCSI server

4. A *controller description and device description* of the form NWSDNET and NWSDTCP. We do not describe these in more detail, because there are no changes and they are not relevant to the discussion.
5. To display the default objects associated with the Integrated servers, use the command WRKCFGSTS *NWS as in Figure 4-7 on page 58.

```

Work with Configuration Status                                ITCDEM02
                                                            02/26/06 16:09:08
Position to . . . . . Starting characters

Type options, press Enter.
 1=Vary on   2=Vary off   5=Work with job   8=Work with description
 9=Display mode status   13=Work with APPN status...

Opt  Description      Status      -----Job-----
    ANWSD             VARIED OFF
    ANWSDV3           VARIED OFF
    ANWSDPP           VARIED OFF
    ANWSDNET          VARIED OFF
    ANWSDTCP          VARIED OFF

```

Figure 4-7 Displaying default objects created by the install

```

Display Network Server Desc                                ITCDEM02
                                                            02/21/06 17:44:51

Network server description . . . . : ANWSD
Option . . . . . : *VRTETHPTH
Virtual Ethernet path . . . . . :

-----Host adapter VE paths-----
Port      Host      Path
number    name      resource
*VRTETHPTH AS300NWSH1    CMN09

```

Figure 4-8 Illustration of new Virtual Ethernet path in NWSD

- Virtual disks (*NWSSTG*), which are a *System drive* and an *Install Source* drive. These are files created from the System i5 storage, which appear to the Windows operating system as the local C and D drives. After the install, additional user disks can be created and added dynamically to the server. When the disks are created and linked, they appear to Windows as new disks that need to be formatted. To work with the disks, use the *WRKNWSSTG* command. Alternatively, you can use Series Navigator to work with the disks. To use iSeries Navigator to work with disks, expand Integrated Server Administration and click **All Virtual Disks**.

Note: iSCSI does not support fixed linking of virtual disks.

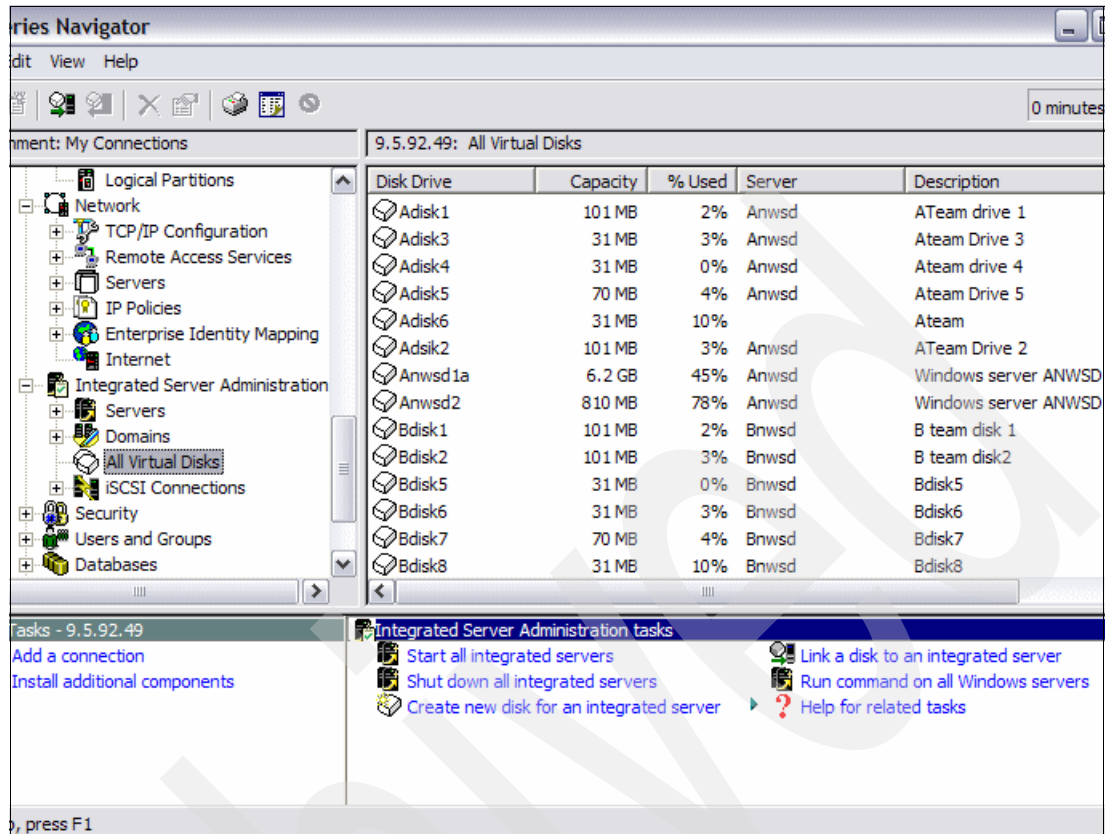


Figure 4-9 Displaying virtual disks on iSeries

4.2.2 New configuration objects required for iSCSI

Now we discuss new configuration objects required for iSCSI.

The Network Server Host Adapter (NWSH)

There are several new configuration objects required for iSCSI servers. One of them cannot be created by the install command (NWSH) and the other three are new objects of type *NWSCFG. Understanding what is configured in the NWSH and NWSCFG objects is the key to being able to utilize the iSCSI architecture to its fullest. These objects were discussed in Chapter 1, "Introduction to iSCSI integrated server support" on page 1 and Chapter 3, "Planning for iSCSI attached servers" on page 43.

To create an NWSH using iSeries Navigator, use the following steps:

1. Expand **Configuration and Service** → Expand **Hardware** → Click **Communications** (Figure 4-10).

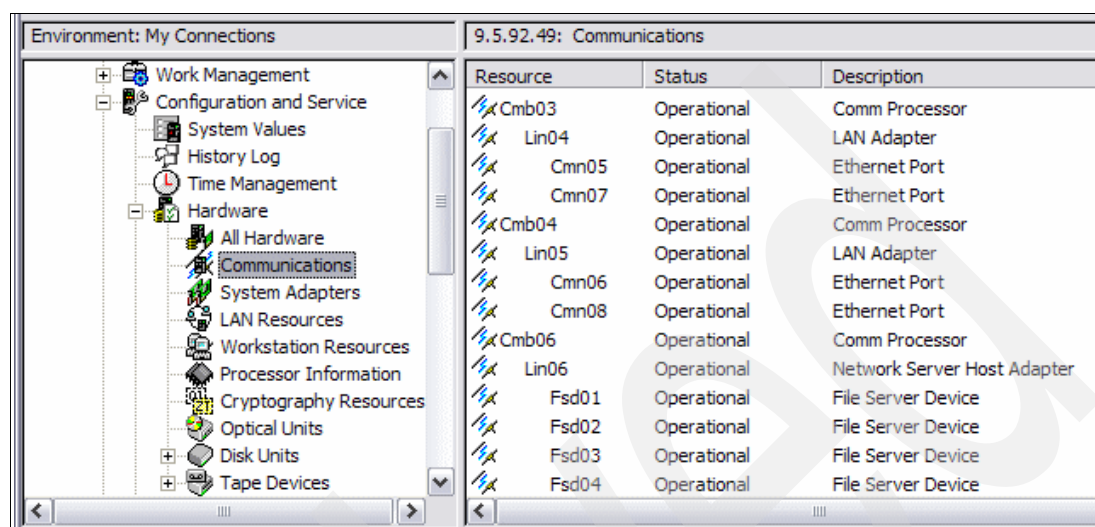


Figure 4-10 Locating NWSH hardware

2. Select each LINXX resource described as a Network Server Host Adapter (Figure 4-11).

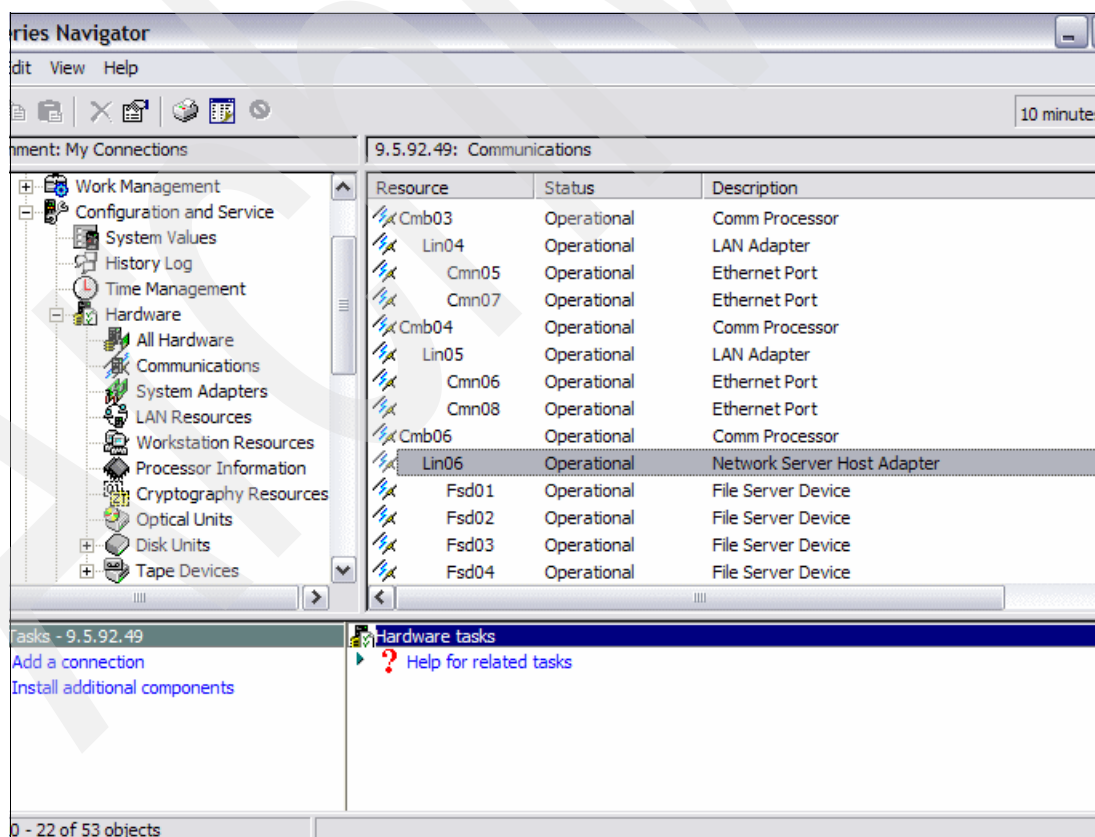


Figure 4-11 Select Network Server Host Adapter

3. If there are more than one adapter, right-click each **Network Host Adapter Resource** → **Properties** → **Physical Location** and note the Frame ID and Card to determine which adapter you will configure (Figure 4-12).
4. Note the LINXX value for that resource.

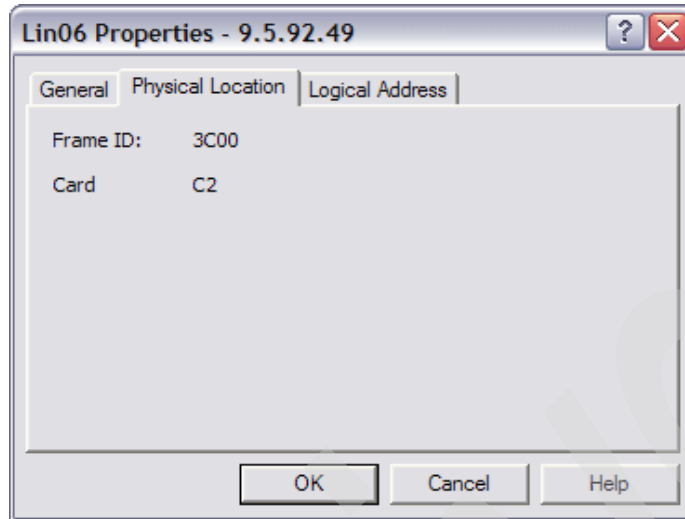


Figure 4-12 Verifying which adapter to configure

5. Expand **Integrated Server Administration** → **iSCSI Connections** → **Select Local Host Adapters** (Figure 4-13).

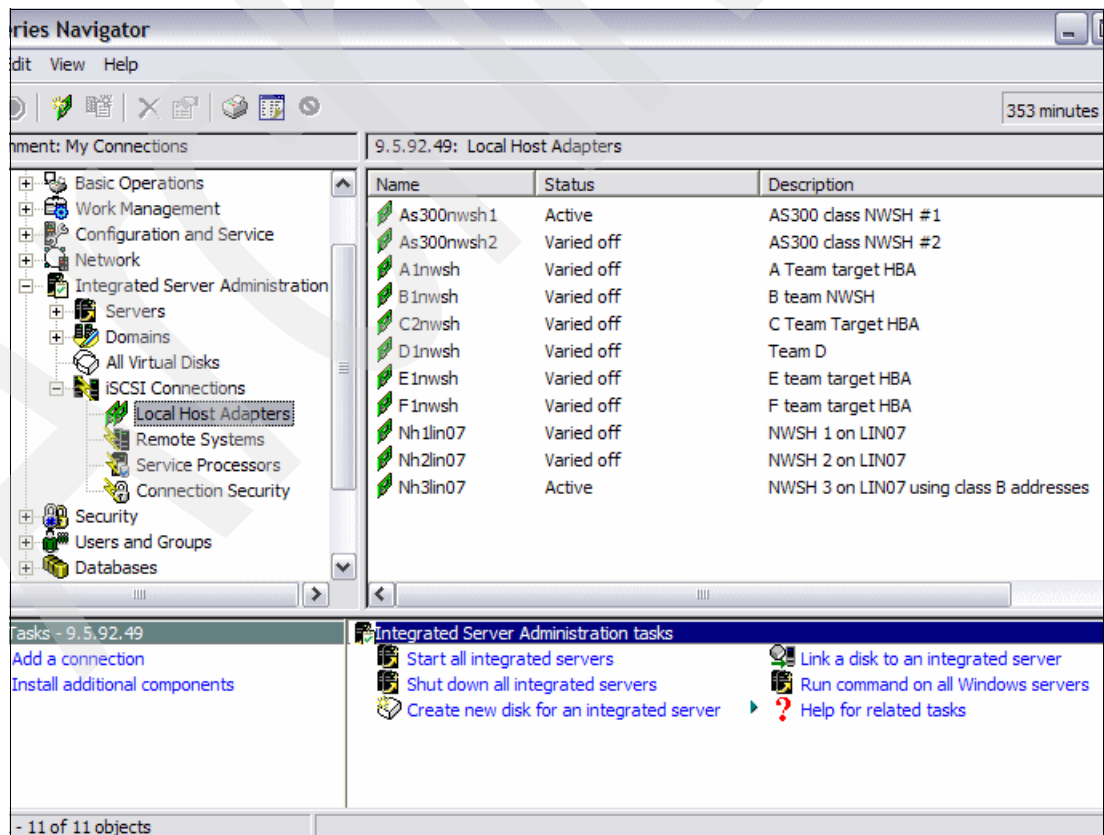


Figure 4-13 Select Local Host Adapter

6. Right-click **Local Host Adapter** and click **New Network Host Adapter** (Figure 4-14).

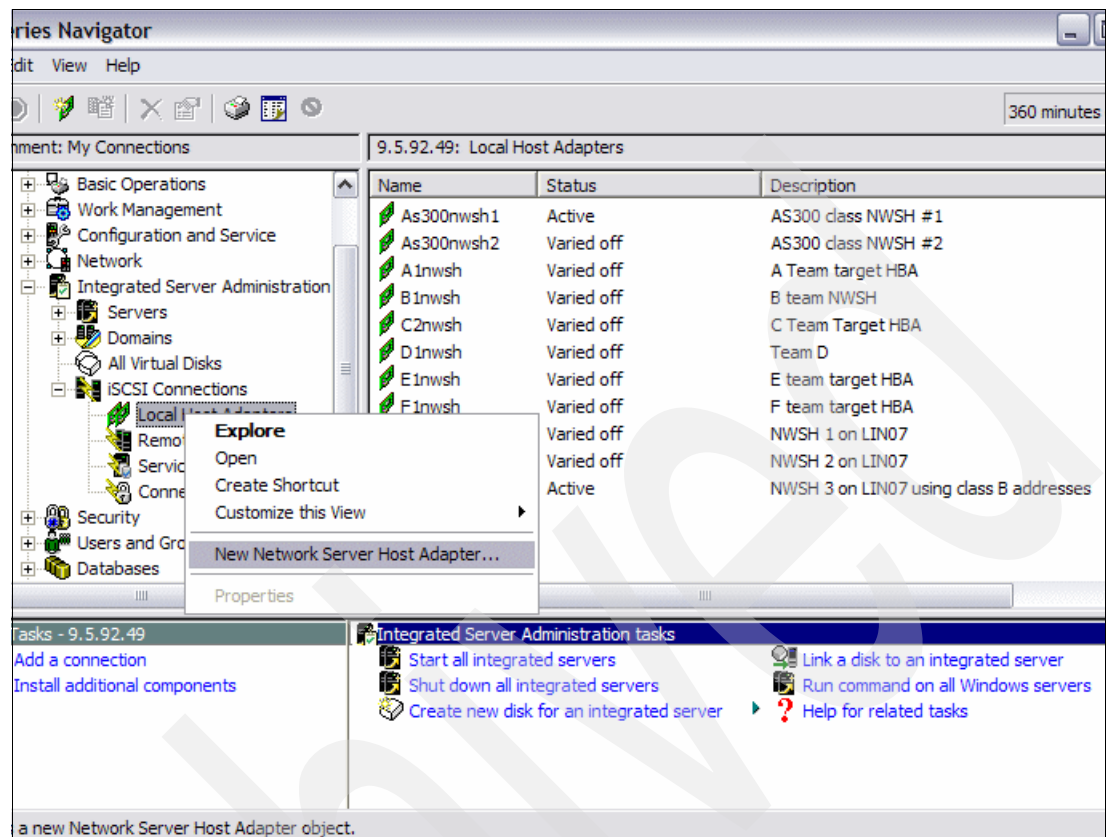


Figure 4-14 Select New Network Host Adapter to create new NWSH

7. Provide a name using an appropriate naming convention. Fill in the Resource name derived in Figure 4-10 on page 60, choose whether to have the NWSH online at IPL and specify the authority for users not explicitly authorized to the object (Figure 4-15).

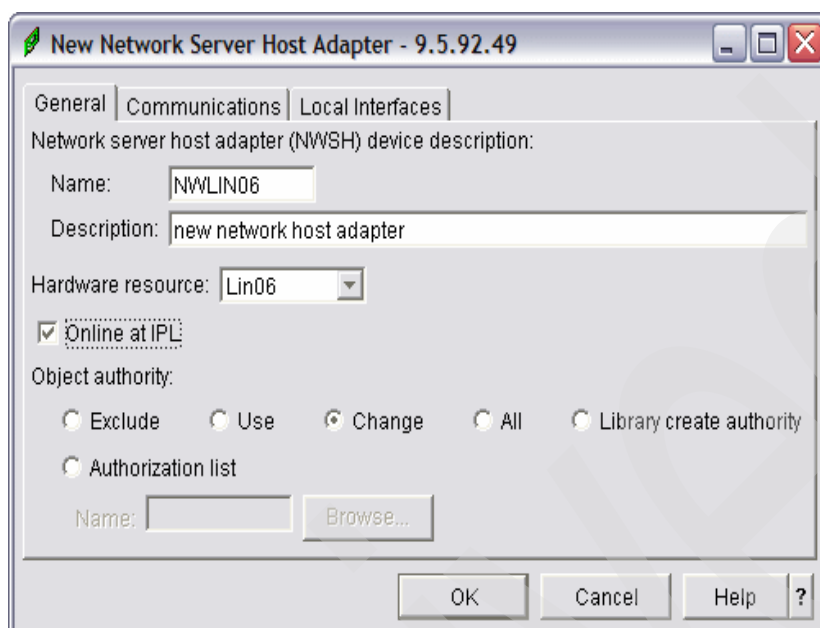


Figure 4-15 Specify hardware resource for network host adapter

8. Fill in the communications page (Figure 4-16). Using the defaults should be fine, although you might want a specific message queue for communications messages.

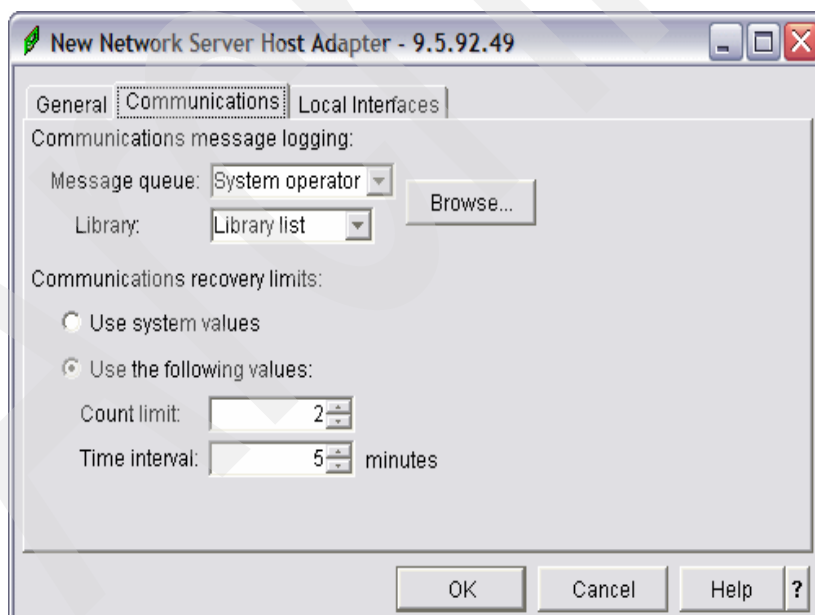


Figure 4-16 Communications tab of new NWSH

Select **Local Interfaces** and fill in addressing for the SCSI and Lan interfaces of the iSeries HBA. These addresses should have been determined previously using section 3.4.3 of the *iSCSI Network Planning Guide* in the iSCSI install readme first using the addressing schema in that guide or one that makes sense in your network. An alternate schema is presented later in the document, which utilizes a class B addressing scheme to provide more scalability than the sample in the readme.

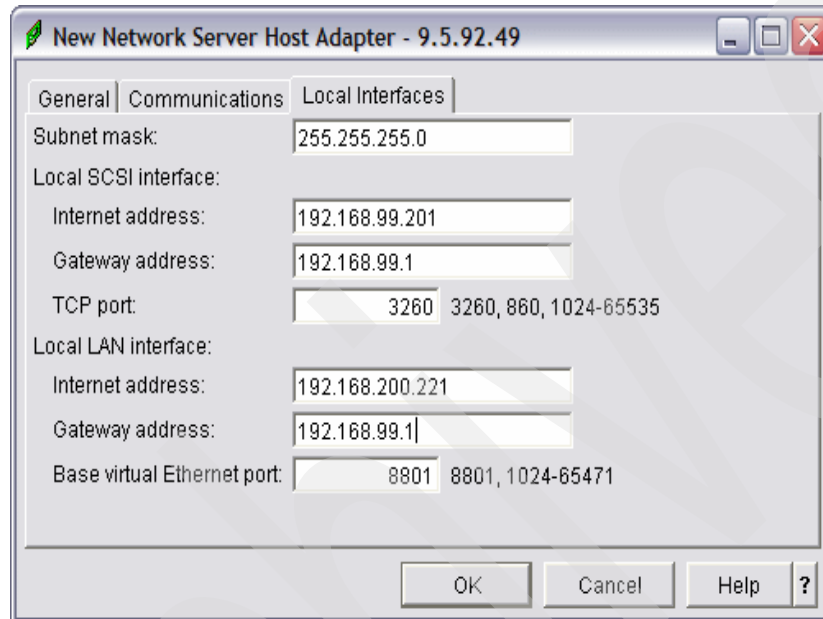


Figure 4-17 Local interfaces for iSeries target adapter.

9. Click **OK**.
10. Right-click the NWSH object that was created and click **Start**.

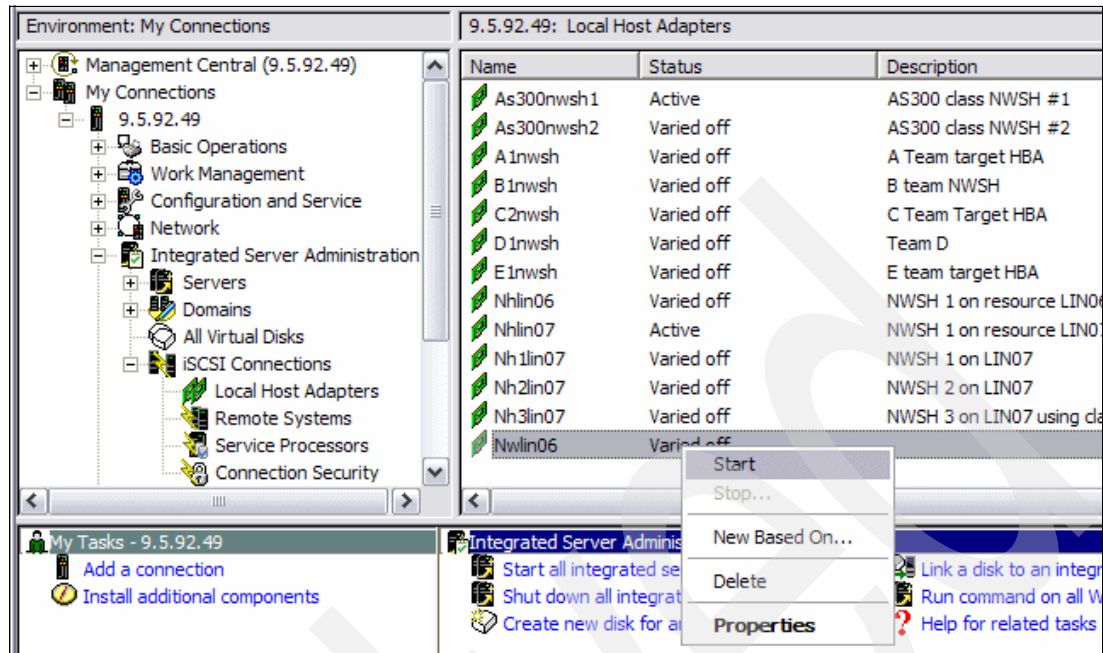


Figure 4-18 Starting NWSH

Note: Notice that you alternatively use `WRKHDWRSC *CMN` to determine the location and resource that you will use and the `CRTDEVNWSH` command as noted previously. This command has two extra parameters: port speed and duplex. However, they only have one parameter so there is really no additional configuration. To work with the description from a 5250 session after it has been created, use the command `WRKDEVD DEVD(*NWSH)`.

4.2.3 The network server configuration objects

The `NWSCFG` objects provide the information needed to discover the service processor and to communicate with the server on the iSCSI network. Also, they can simplify the configuration by using the *integrated DHCP server* to provide some boot parameters to the device. The objects have a type of `*NWSCFG` (network server configuration). There are three subtype descriptions that are required for each server. You can create them using the `CRTNWSCFG` command, but we recommend that you use iSeries Navigator to create each of them.

Service processor configuration object is a type of `*SRVPRC`. It provides the information necessary for IBM Director Server to discover the processor for powering on and off the device.

There is one of these descriptions per xSeries server and one per BladeCenter. The object for the BladeCenter references the Management Module and not the individual service processor on each blade. The object has been further defined in Chapter 1, "Introduction to iSCSI integrated server support" on page 1 and Chapter 3, "Planning for iSCSI attached servers" on page 43. To create a service processor object:

1. Expand the **Integrated Server Administration** → **iSCSI Connections** and highlight **Service Processor**. Figure 4-19 on page 66 shows this.
2. Right-click and specify **New Service processor configuration** (Figure 4-20 on page 67).
3. Fill in a name using whatever convention you have chosen.

4. Check **Use service processor connection to determine remote system enclosure identity**.
5. Put in the IP address of the service processor and the serial number of the enclosure. Figure 4-21 on page 68 shows this.
6. Choose what to make the default authority to the object. See Figure 4-21 on page 68.
7. Take the default on the Security tab.
8. Click **OK**.
9. Initialize the service processor. For information about initializing the processor, see 6.5, "Manage service processor server configuration" on page 169.

Remote system configuration object is a type of *RMTSYS. The remote system describes each initiator machine. There is one of these objects per xSeries server, however, unlike the service processor object, which is shared by all the blades, there is one Remote System object for each Blade server to be installed. To create a remote system object (Figure 4-19):

1. Expand **Integrate Server Support** → **iSCSI Connections** → Highlight **Remote Systems** and right-click it.

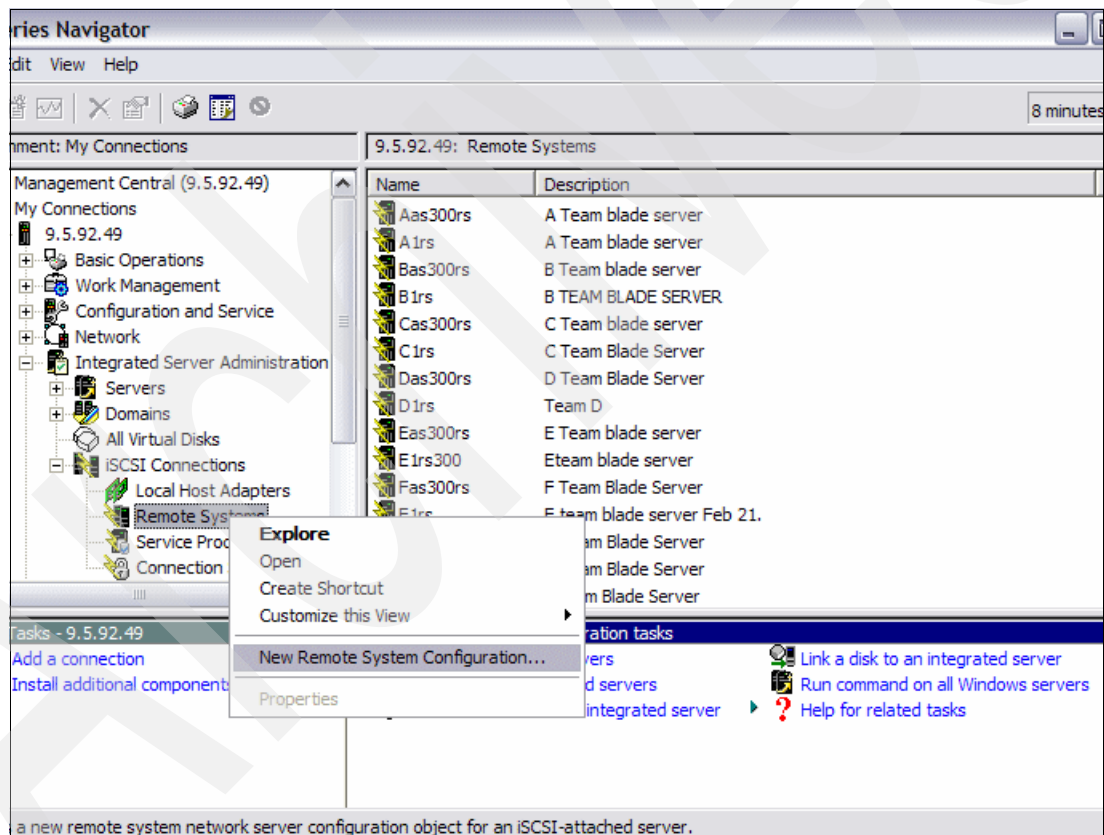


Figure 4-19 Select New Remote System Configuration

2. Click **New Remote System Configuration**.

3. Fill in a name using an appropriate convention.
4. Select the appropriate Service Processor Configuration.
5. For Remote Identity, select **Use the following values** and input the serial number, manufacturer type, and model of the system (Figure 4-20).
6. Select the default Object authority for the object.

New Remote System Configuration - 9.5.92.49

General | Boot Parameters | CHAP Authentication | Network Interfaces

Remote system network server configuration:

Name:

Description:

Service processor configuration:

Remote system identity:

☐ Use enclosure identity from service processor configuration

☒ Use the following values:

Serial number:

Manufacturer type and model:

Object authority:

☐ Exclude ☐ Use ☒ Change ☐ All ☐ Library create authority

☒ Authorization list

Name:

Figure 4-20 Describing the remote system: A Blade in this case

7. On the Boot Parameters tab (Figure 4-21), we recommend that you select **DHCP** and use the default for the Vendor and Client ID fields. Otherwise, identify the boot device using information gathered from the Fast!Util utility.

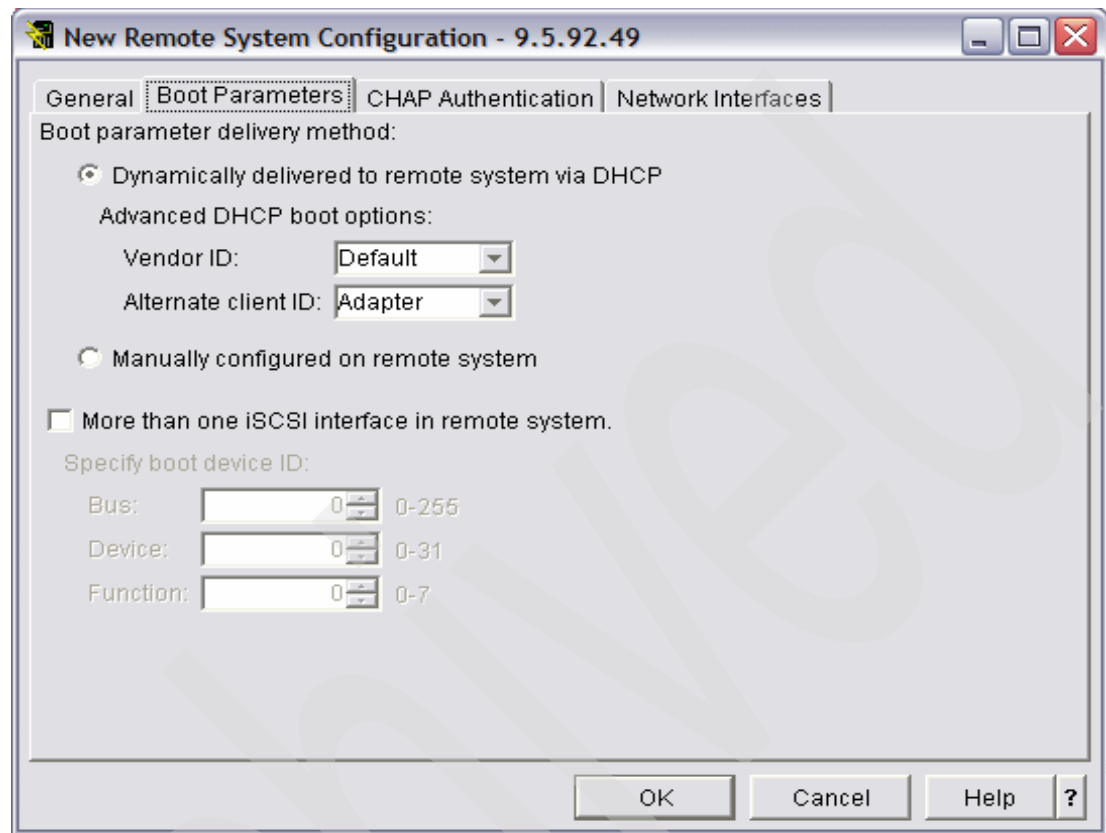


Figure 4-21 Select DHCP for boot parameters

8. Configure CHAP authentication (recommended) in Figure 4-22. If you have already configured the CHAP name and secret on the initiator, specify the same name and password here. If you generate the secret, then it will be a very strong secret, but it will need to be entered carefully on the initiator. We are using CHAP as an example. This is not generally an acceptable CHAP secret. Note also this is only authentication using a secret defined between initiator and target; it is not encrypted.

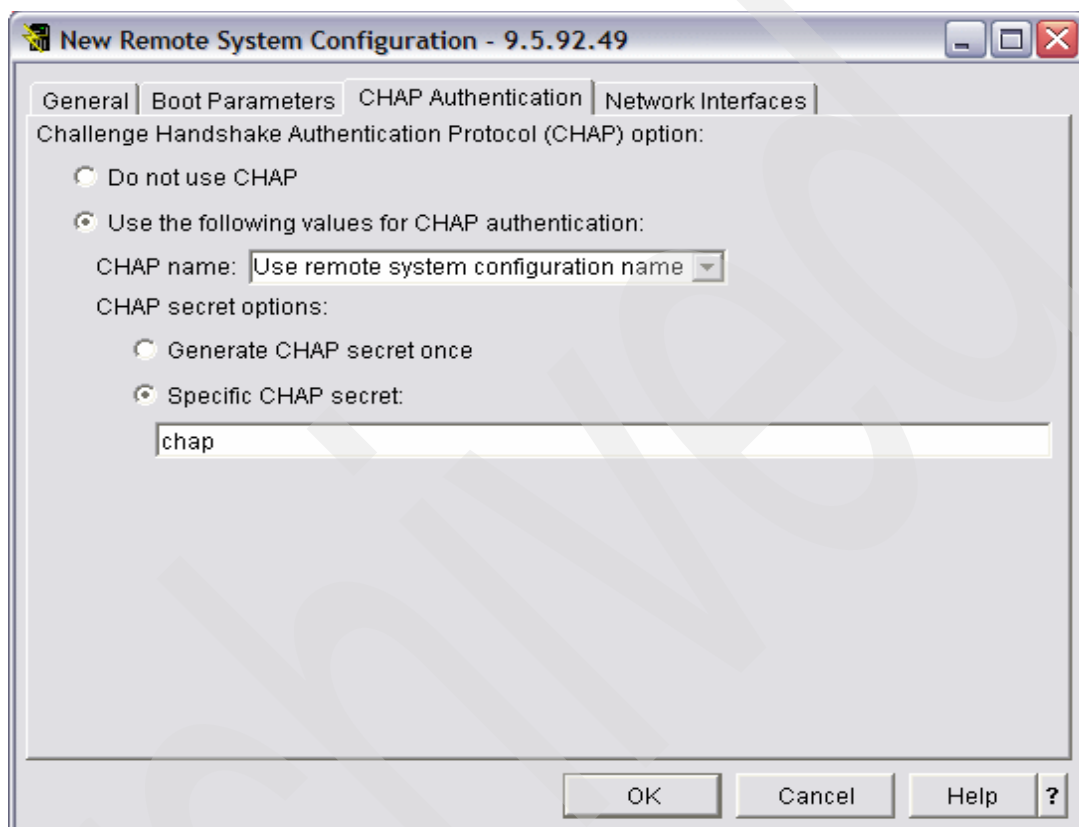


Figure 4-22 Configuring CHAP as part of remote system configuration

9. In the Network Interface Properties (Figure 4-23), input the correct addresses for the SCSI and Lan interfaces on the remote device. If there are multiple ports that you plan to use on the machine, add the same information for each port.

Rskxkj897 Network Interface Properties - 9.5.92.49

Specify configuration values for network interface 1.

Remote SCSI interface:

Local adapter (MAC) address: 00c0dd0759ba

Internet address: 128.168.203.1

Subnet mask: 255.255.255.0

Gateway address: 128.168.200.1

iSCSI qualified name (IQN):

☐ Generate an iSCSI qualified name

☒ Specific iSCSI qualified name:

Remote LAN interface:

Local adapter (MAC) address: 00c0dd0759b9

Internet address: 128.168.204.1

Subnet mask: 255.255.255.0

Gateway address: 128.168.200.1

OK Cancel Help ?

Figure 4-23 Configuring remote interfaces in the remote system configuration object

10. Click **OK**.

11. To verify the configuration, check the status. To check the status of the remote system using iSeries Navigator, do the following:
 - a. Expand **Integrated Server Support** → **iSCSI Connections** → **Remote Systems**
 - b. Select the remote system object you wish to verify on Figure 4-24.

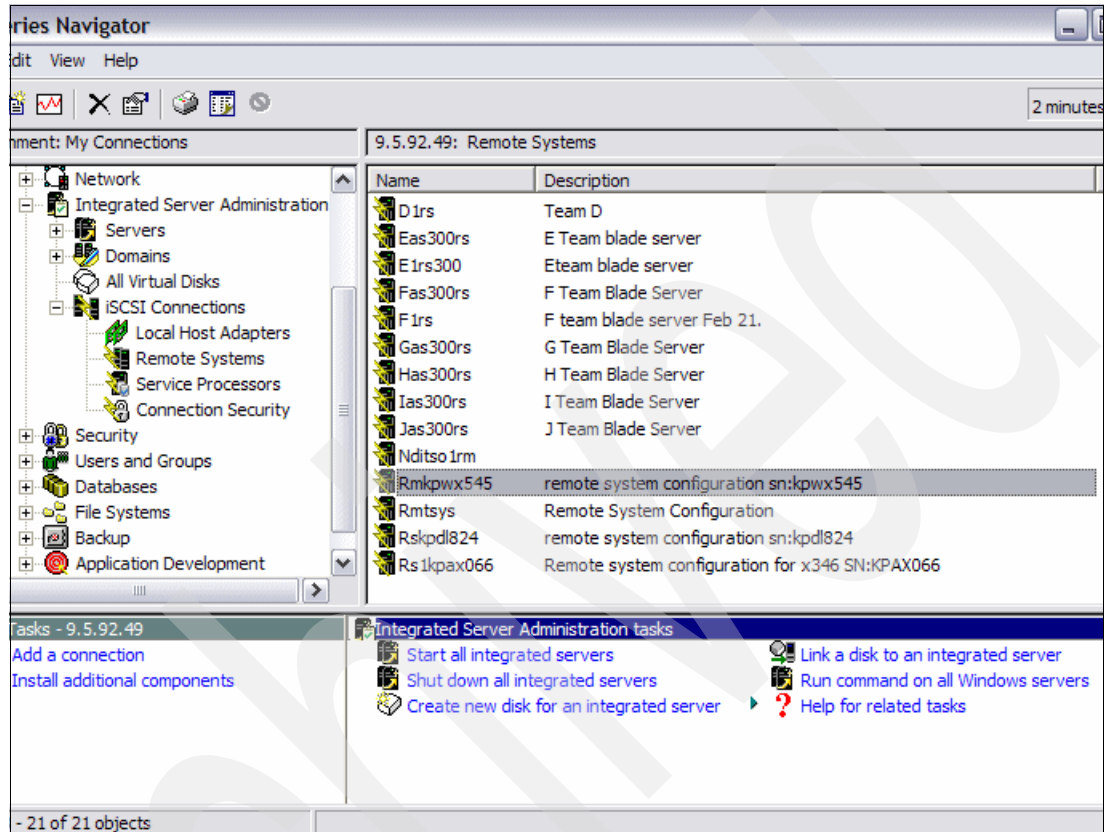


Figure 4-24 Selecting the remote system to verify

- c. Right-click the remote system and select **Status** (Figure 4-25).

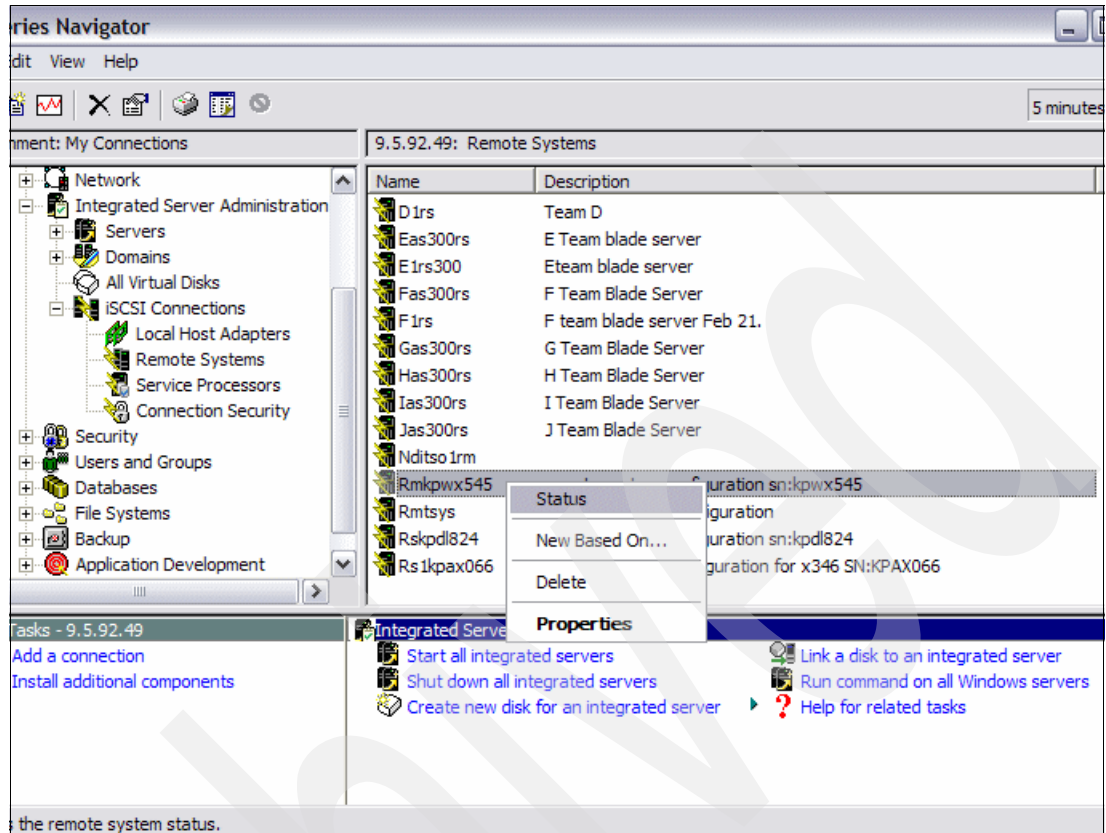


Figure 4-25 Select status

- Click **Status** and the window returns with status. In Figure 4-26, it finds the server is powered on confirming that the server can be found. If the server was powered off but could still be found, it would return a status of powered off (Figure 4-27 on page 73). Another status that might need to be investigated is listed in Figure 4-28 on page 73.

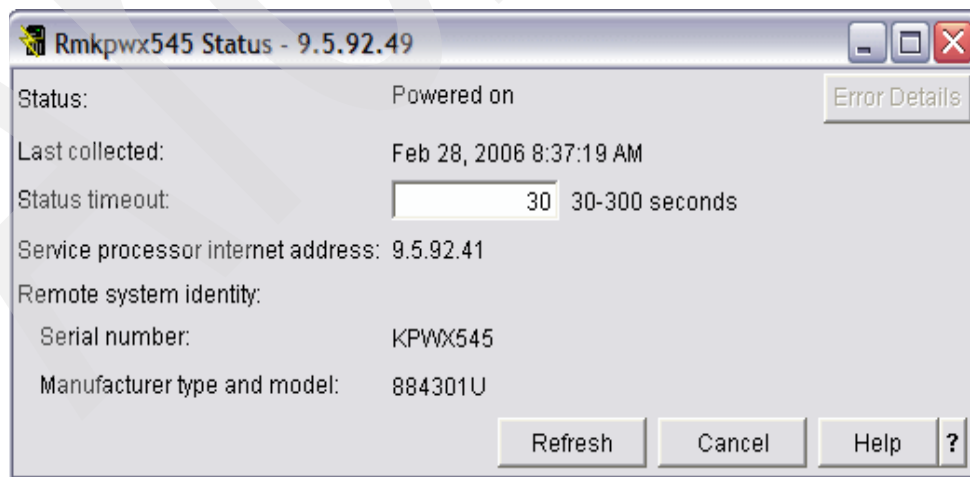


Figure 4-26 Verification that Director can find the server

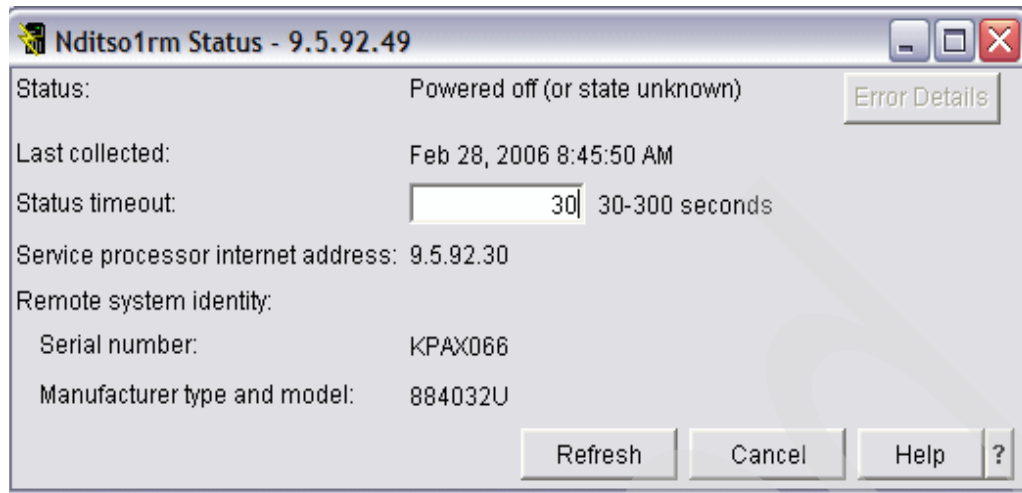


Figure 4-27 Remote system is powered off or otherwise unavailable

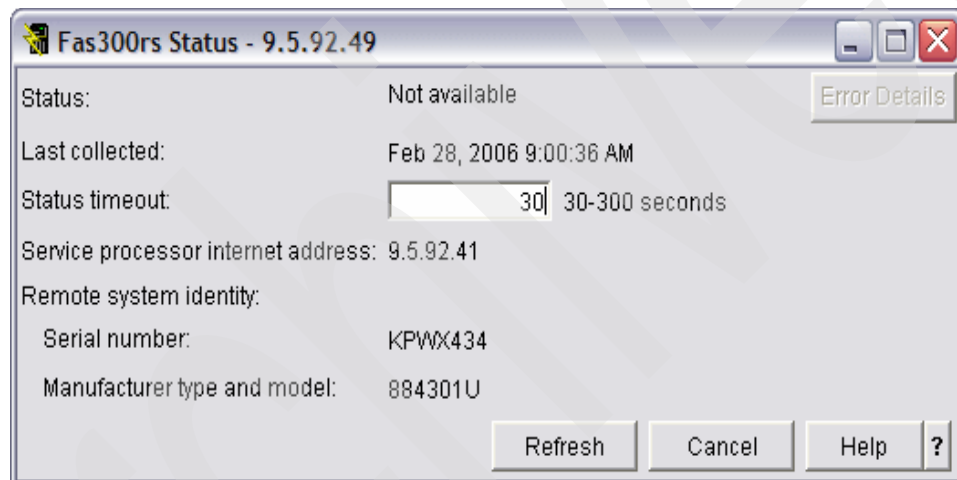


Figure 4-28 Verifying remote system that is unavailable

- d. Alternatively, you can verify the status of the remote system by using the WRKNWSCFG command and selecting option 8. See Figure 4-29 on page 74.

Work with NWS Configuration

System:

Type options below, then press Enter.

1=Create 2=Change 4=Delete 5=Display 6=Print 8=Work with status

Opt	Name	Type	Text
	JAS300RS	*RMTSYS	J Team Blade Server
	MMSP	*SRVPRC	Management Module in the BladeCenter
	NDITS01CN	*CNNSEC	
	NDITS01RM	*RMTSYS	
	NDITS01SP	*SRVPRC	
	RMKPWX545	*RMTSYS	remote system configuration sn:kpwx545
	RMTSYS	*RMTSYS	Remote System Configuration
	RSKPDL824	*RMTSYS	remote system configuration sn:kpd1824
8	RS1KPAX066	*RMTSYS	Remote system configuration for x346 SN:KPAX066
	SPKPKY358	*SRVPRC	Management Module configuration sn:KPKY358
	SP1KPAX066	*SRVPRC	Service processor configuration x346 SN:KPAX066

Bottom

Parameters or command

==>

Figure 4-29 Using 5250 screen to verify remote system status

Connection security configuration object is a type of *CNNSEC. This object is required but not used at present. Thus, one object could be used for all servers. In the future, it could be used to define security options on a per server basis. To create a connection security object:

1. Expand **Integrated Server Support** → **iSCSI Connections** → right-click **Connection Security**. See Figure 4-30.

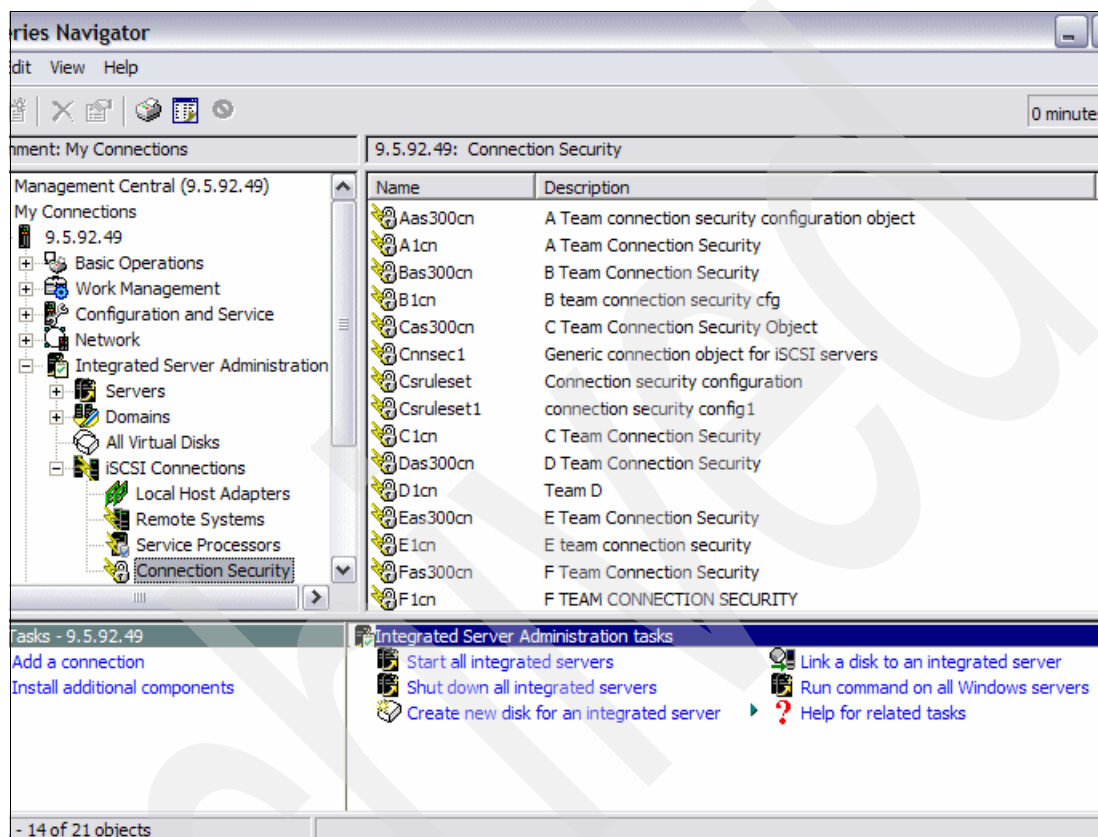


Figure 4-30 Select Connection Security object

- e. Click **New Connection Security Configuration**. See Figure 4-31 on page 76.

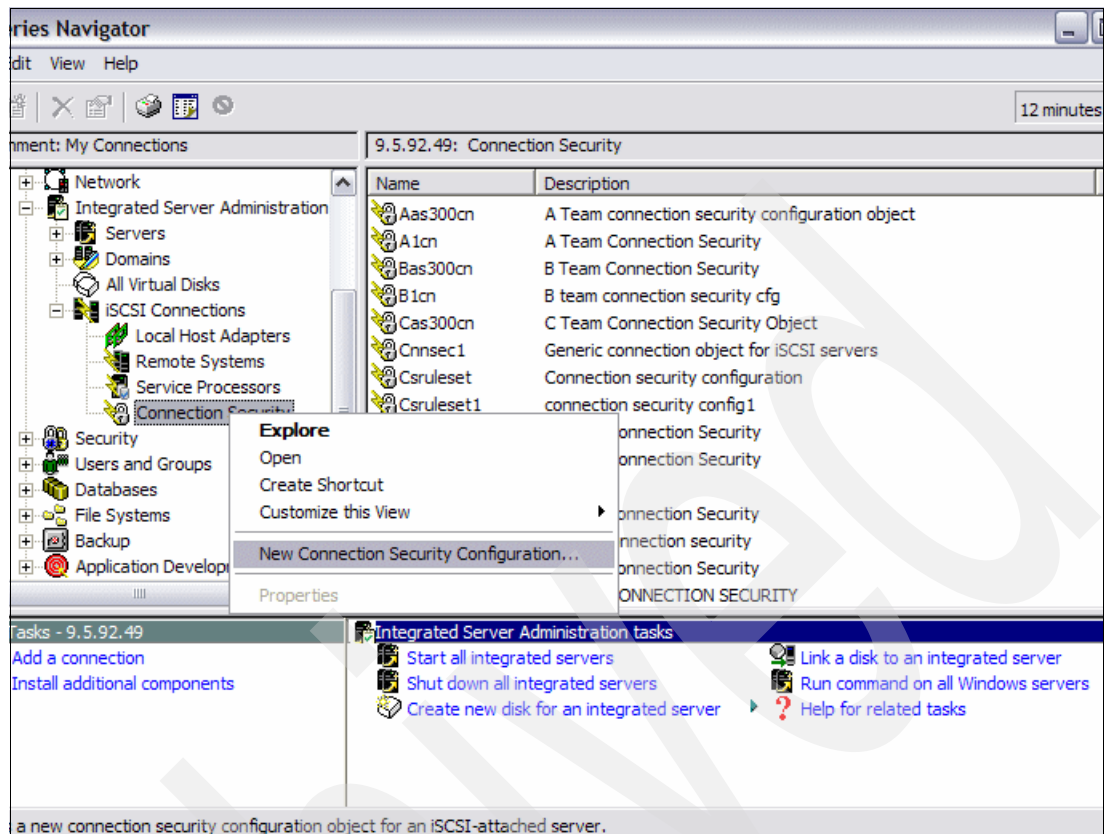


Figure 4-31 New Connection Security Object

- f. Assign it a name.
- g. Specify the default Object Authority.
- h. Ignore IP Security Rules. See Figure 4-32.

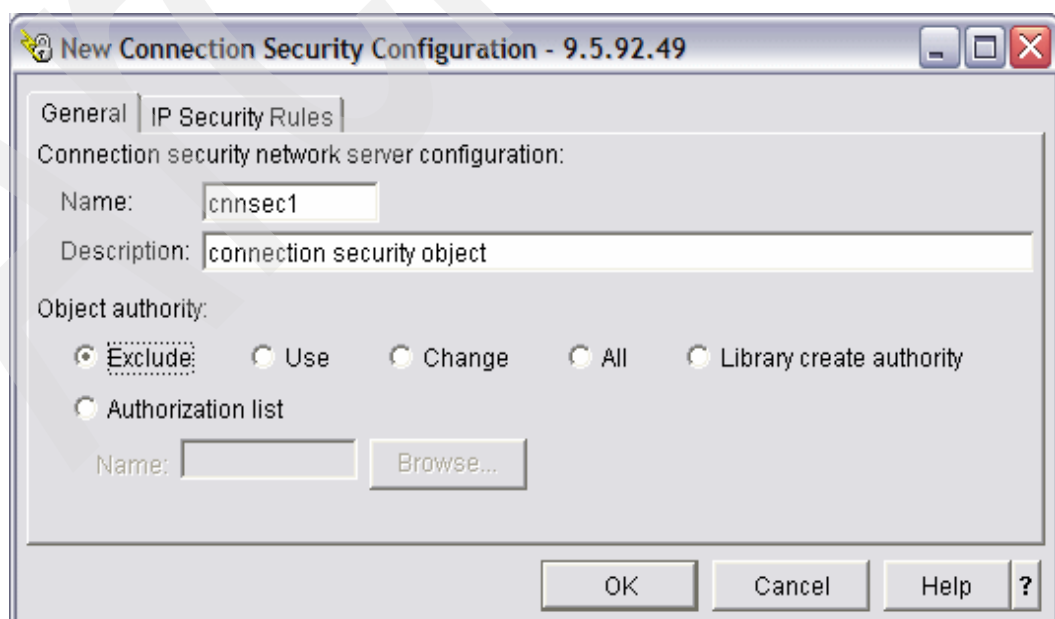


Figure 4-32 Defining a connection security configuration object

- i. Click **OK**.

4.2.4 Reasons for precreating *nwscfg objects prior to the install

The install command can create the *NWSCFG objects, but there are several reasons to precreate the objects:

- ▶ It simplifies the install considerably. When the objects are not precreated, there are many more entries that need to be made in the command so that it is very easy to make an error.
- ▶ It does not provide a prompt for multiple remote interface entries, which would mean that if you had multiple ports, that is, multiple HBA adapters in your xSeries, or both ports enabled on the blade, you would have to go back and add these entries to the remote system object after the install using iSeries Navigator.

Note: At the time of this writing, the CRTNWSCFG and WRKNWSCFG interfaces have a functional restriction, which prevents the prompter from adding additional entries for the remote interface. Use iSeries Navigator to accomplish this.

- ▶ It improves the chances that the install will not encounter unanticipated communications problems, because initializing the processor configuration object (*SRVPRC) verifies that this communication is working and requesting status on the remote system (*RMTSYS) verifies that the specific Blade or xSeries can be communicated to also.
- ▶ The install command does not allow you to customize the names of the *NWSCFG objects to fit your naming convention.
- ▶ It works nicely with using the network planning guide worksheet, particularly in an environment in which you will be using initiator machines with multiple iSCSI sessions.
- ▶ If you have multiple servers to install, it definitely speeds up the process. You precreate all the objects needed and then plug them in to each command. This facilitates running multiple install commands simultaneously.

4.2.5 Install types

The install command lists two types of install: *FULL and *BASIC. The *BASIC type install was added when the IXA support was added. It allowed part of the install to proceed from the xSeries server using the Server guide software in order to simplify the installation of drivers for xSeries. This option is not allowed at this time. The only valid install type for iSCSI at this time is *FULL. There are places in the documentation that do not make this clear.

Important: The *BASIC install type is not supported at this time when running INSWNTSVR to install an iSCSI server.

4.2.6 Removing System i5 objects

If something goes wrong with the install, or for some reason you need to clean up the objects that the install created, then you issue the command:

```
DLTWNTSVR NWSD(NWSDNAME)
```

Doing so will delete the NWSD, Virtual Ethernet lines, the controller, and device that are created by the install command, the system, and install source drives, and the TCP/IP interface that is created for the System i5 side of the Virtual Point-to-Point line description.

The command can be used at any time, but after the install there are other objects that you need to remove if you wanted to remove the server completely. The command does not remove any of the other objects associated with the server such as the NWSH, and NWSCFG objects, nor does it remove any additional storage spaces that might have been created after the install. The additional objects can all be deleted using iSeries Navigator, or by using the following commands:

- ▶ WRKDEVD DEVD(NWSH) and taking the option to delete.
- ▶ DLTNWSCFG and specifying the name of the configuration object.
- ▶ RMVNWSSTGL to unlink the space and then DLTNWSSTG.

4.2.7 Installation

As was stated in the beginning, the install command installs a single instance of Windows Server 2003; however, there are multiple approaches that you can take based on the amount.

Additional Considerations

We also want you to consider the following:

- ▶ We recommend installing using the precreated objects as noted previously in 4.2.4, "Reasons for precreating *nwscfg objects prior to the install" on page 77.
- ▶ If you choose to utilize the install command to create the objects, make sure that you thoroughly understand the parameters involved and that you might have additional configuration to do after the fact.
- ▶ There are many parameters that are not required or supported with this install at the time of writing this book. Most notable of these are the parameters referencing clustering (Figure 4-33) and those referencing IPSEC. These functions might become available in the future, so if you are interested you should watch the Integrated server site at:

<http://www-03.ibm.com/servers/eserver/iseries/integratedxseries/windows/>

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Additional Parameters

Keyboard layout	*DEFAULT	Hexadecimal value, *DEFAULT
Configuration file	*NONE	Name, *NONE
Library		Name, *LIBL, *CURLIB
Cluster name	*NONE	Name, *NONE
Cluster configuration:		
Cluster domain name	*CLU	
Quorum resource size		550-1024000, *CALC
Quorum resource ASP		1-255
Quorum ASP device		Name
Connection port		*VRTETH0, *VRTETH1...
Cluster internet address . . .		
Cluster subnet mask		

Figure 4-33 Cluster parameters which should not be configured for iSCSI at this time

- There are many other parameters that can be defaulted and usually we would recommend that those parameters that can be filled in during the Microsoft portion of the install be left until that time and anything else that can be defaulted left as the default in order to simplify the command. If you have questions about a particular parameter, you can probably determine its function and whether you need it or not by moving the cursor onto the parameter and pressing F1 for help.

Windows license key		
License mode:		
License type	*PERSEAT	*PERSEAT, *PERSERVER
Client licenses	*NONE	5-9999, *NONE
Terminal services	*NONE	*NONE, *TSENABLE...

Figure 4-34 Parameters that can be left to the Microsoft portion of the install

Note: The parameter IP security rule has a default of *dftsecrule. Unless you change this to *none, the install will halt.

- The installation advisor referenced in the planning might assist in simplifying what you need to enter for the command.
- We recommend that you configure the remote system boot parameters to use DHCP. This requires that the initiator also be configured to use DHCP using the FastUtil utility, otherwise referred to as *Control Q*.
- If you use CHAP, you want to make sure that the initiator has the correct CHAP name and secret. Alternatively, configure the client but install with the remote system configuration set, so that it does not use and then change this after the install.
- Determine if you need to create additional Virtual Ethernets and if the host adapter that you intend to use is the same as for the iSCSI connection.
- If you are doing an install from the remote control panel of the System Processor's management interface, you might experience a loss of control over the keyboard during the Windows portion of the install. From the management module, check to see that the keyboard icon is highlighted (Figure 4-35 on page 80). If it is not, you might not be able to use the keyboard.

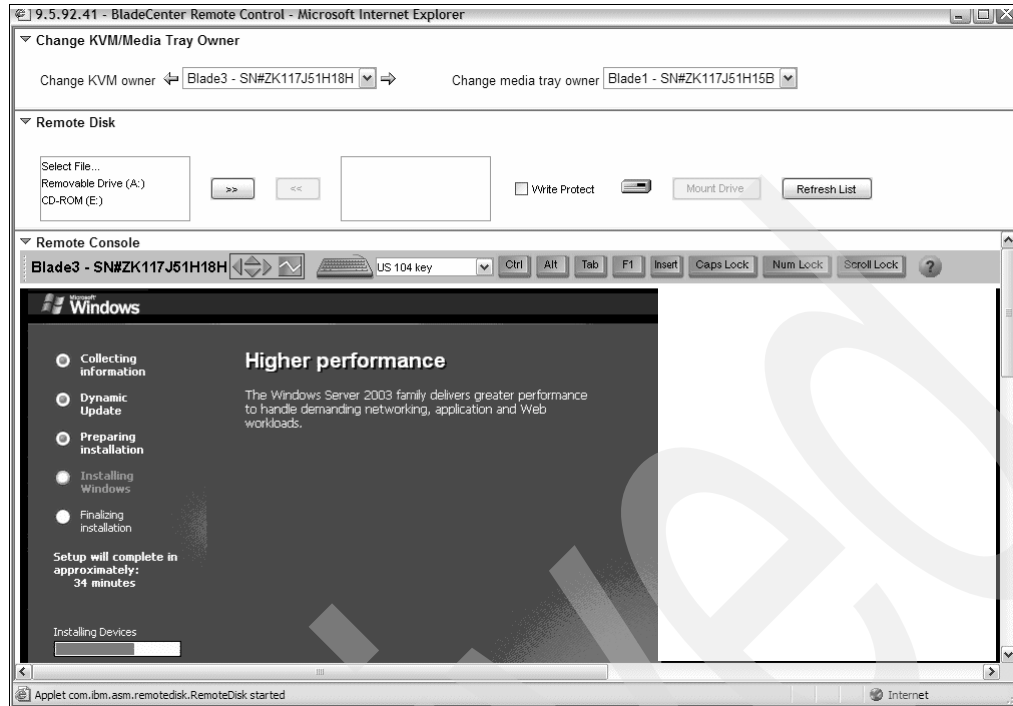


Figure 4-35 Install from the remote control panel verify keyboard icon is highlighted

Installing with the NWSCFG objects precreated

We will document a simple install with only the required parameters configured and using the precreated NWSCFG objects. We will configure an additional Virtual Ethernet line, but we default to everything else that can be defaulted.

Before starting to install the server, verify that:

- The appropriate NWSH is active (Figure 4-36 on page 81). To verify that the appropriate NWSH is active: Expand **Integrated Server Support** → **iSCSI connections** → Double-click **Local Host Adapters** and verify that the correct NWSH is active.

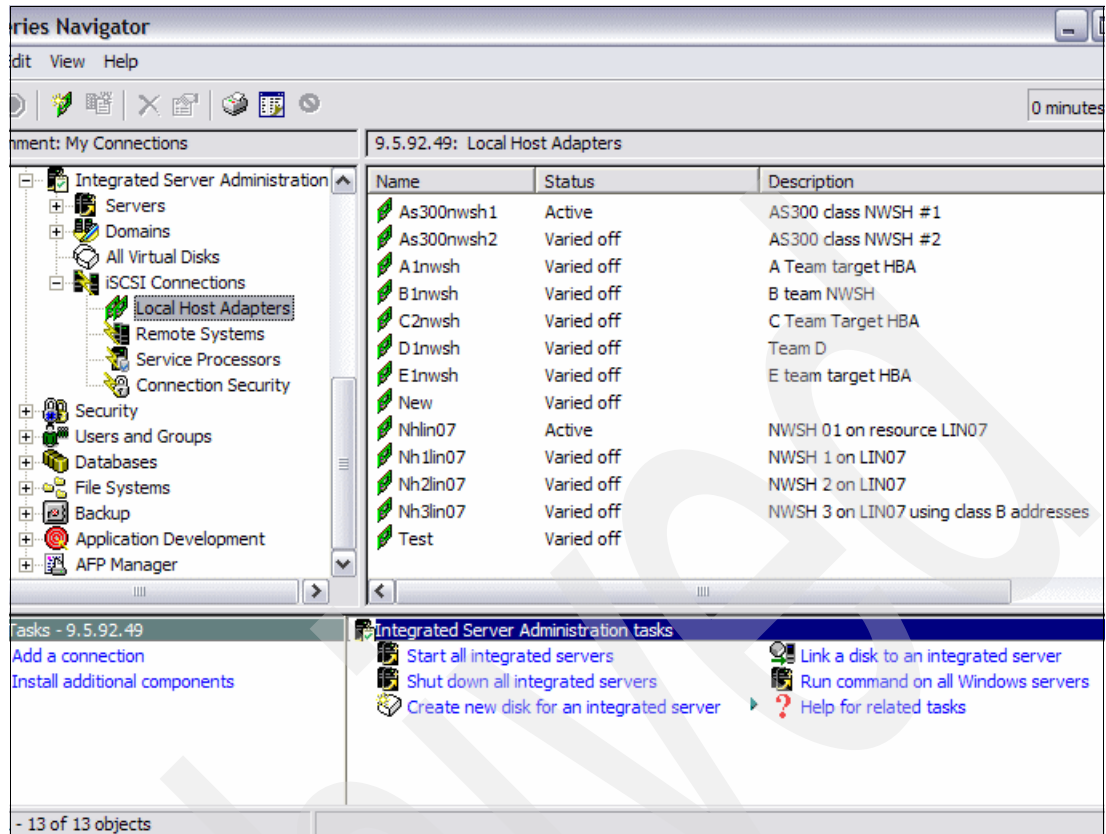


Figure 4-36 Verify that the NWSH is active

- Verify that IBM Director server is started. To verify the status of Director, Expand **Network** → **Servers** → then double-click **User-Defined**.

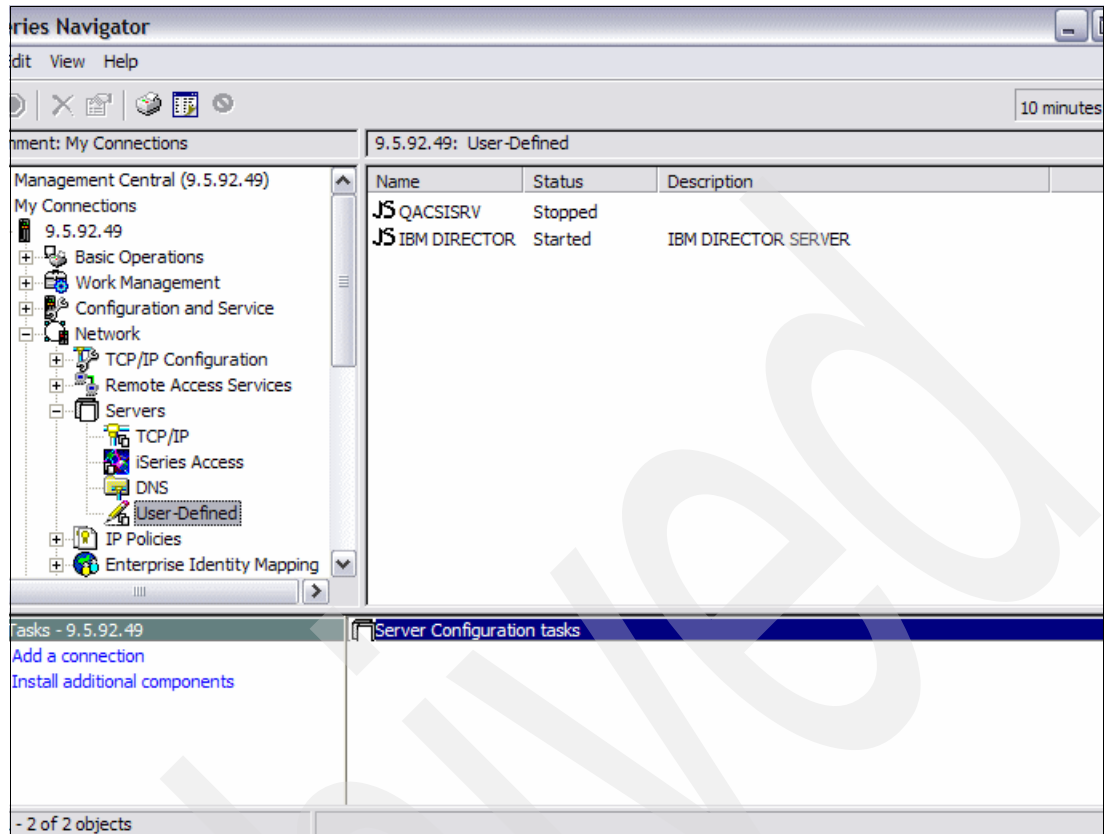


Figure 4-37 Verify the status of IBM Director Server

- The Blade or xSeries server must be powered off to do the install. If you have connected the service processor to a network that is accessible from a browser, you use the browser to access the service processor for xSeries or BladeCenter to verify the status of the server if it is remote from you (Figure 4-38 on page 83).

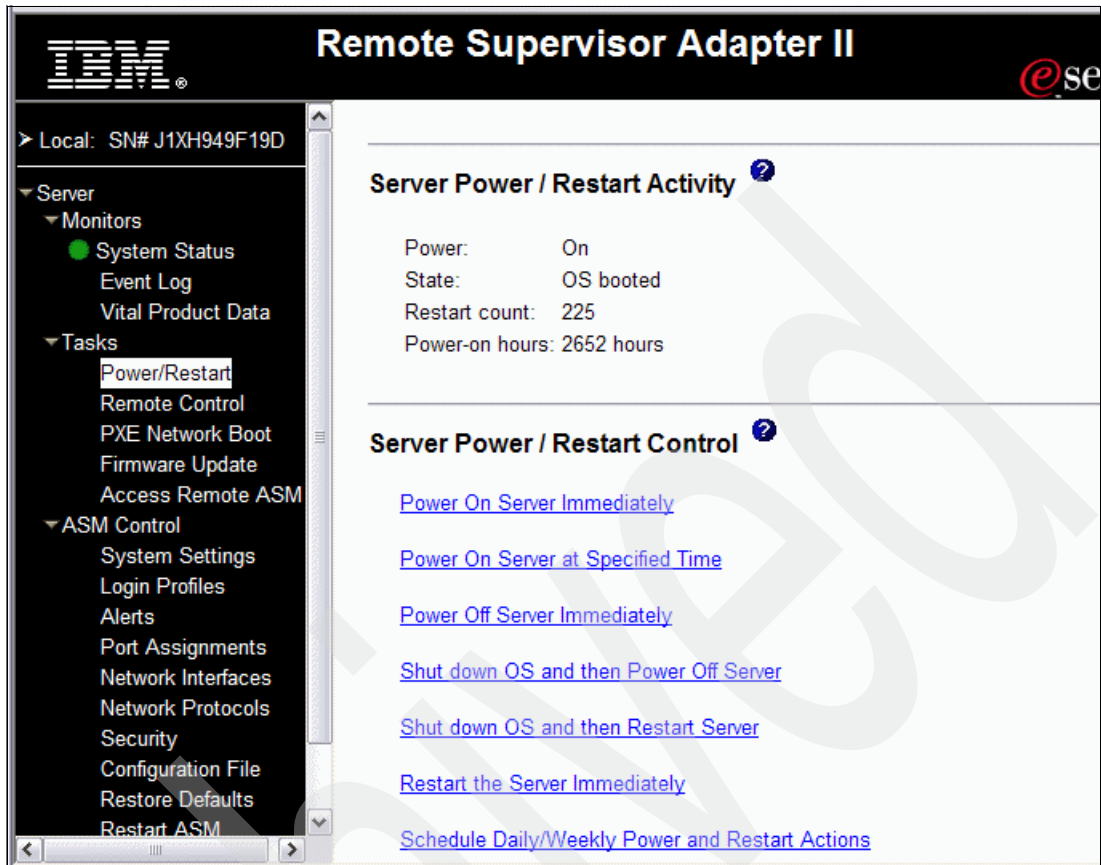


Figure 4-38 Verify status of xSeries server from RSA

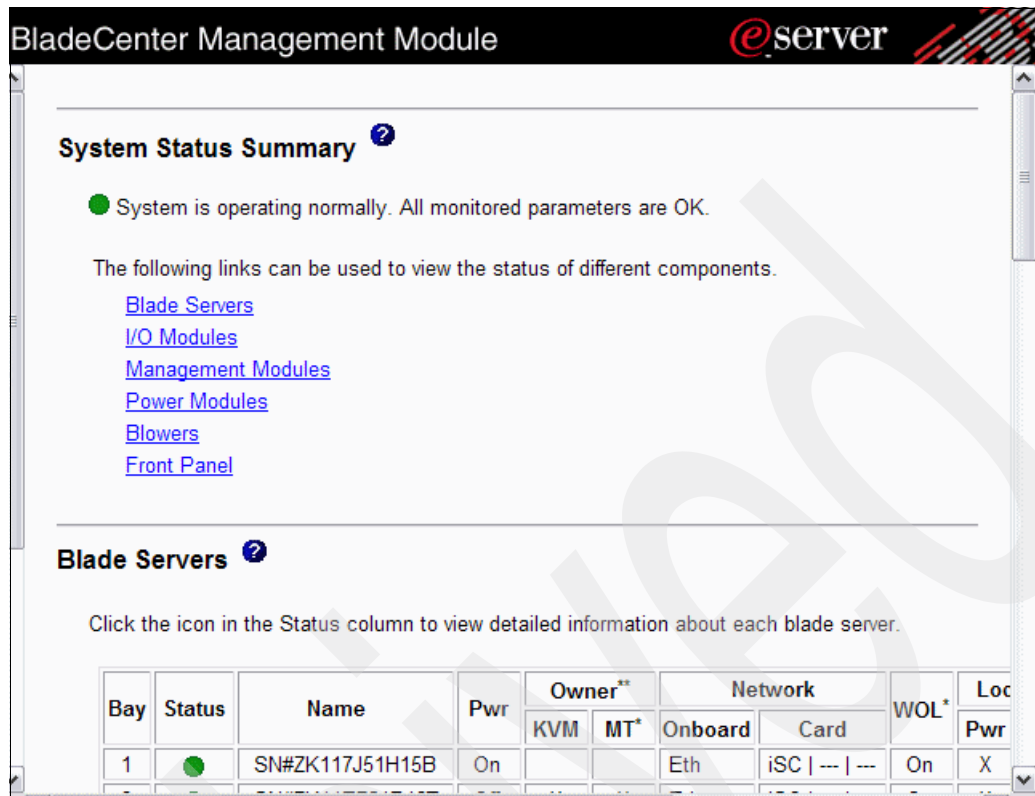


Figure 4-39 Verify status of Blade server

To start the install, go to the command line of a 5250 session (Figure 4-40):

1. Type: INSWNTSVR
2. This produces a window that prompts you for a name. Remember that once created, the name cannot be changed on the System i5.

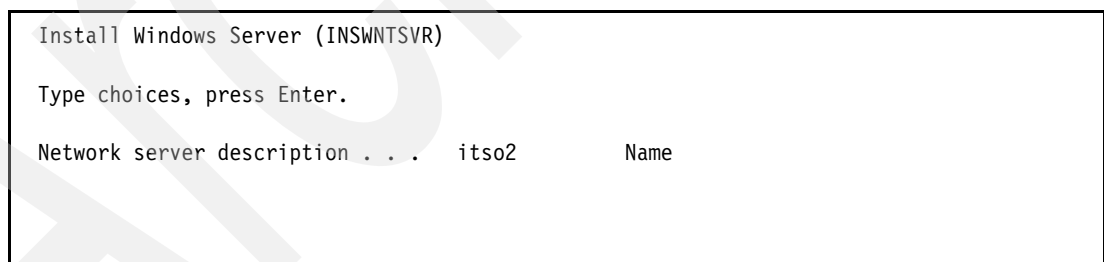


Figure 4-40 First window of INSWNTSVR

3. Press Enter and more parameters are presented. These parameters are required. The installation type has to be `*full`. It is an iSCSI install, and the Windows version has to be `*win2003` (**Windows Server 2003**).

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Network server description . . . > ITS02          Name
Installation type . . . . . *full                *FULL, *BASIC
Resource name . . . . . *iscsi                   Name, *ISCSI
Windows server version . . . . *win2003          *WIN2000, *WIN2003

```

Figure 4-41 Window two of INSWNTSVR

4. Press Enter again and the third window appears. The install source parameter is the path to the Windows install image. If you are installing from the System i5 optical drive, `*dft` is fine. Checking the Help text for that line will document that `*dft` looks in `/QOPT`.
5. If you copied the image into an IFS (Integrated File System) folder to do multiple installs or to merge the SP1 code with the install image, you would define the IFS path from the root to the level that the I386 directory resides in (Figure 4-42). In our case, we had I386 in the `/cdimages/ws2003` directory and that is the path we would have put in here. `*Install` is the only valid entry for iSCSI at this time. We default the other entries and leave that configuration to perform in the Windows portion of the install.

```

Windows source directory . . . . *DFT

Install option . . . . . *INSTALL                *INSTALL, *UPGRADE
TCP/IP port configuration:
  Port . . . . . *NONE                          *NONE, 1, 2, 3, 4
  Windows internet address . . .
  Windows subnet mask . . . .
  Windows gateway address . . .
    + for more values

```

Figure 4-42 Window three of INSWNTSVR

6. This window has a prompt for more entries, so page down. You can configure additional Virtual Ethernet connections on the next window, Figure 4-43. This will create a Virtual Ethernet line description with the NWSD name and a VX appended where x is the port number selected, which in our example would be v1 based on the selection *VRTETH1. It also creates the Windows Lan connection.
7. A parameter of interest is the associated port, which is for doing inter-LPAR Virtual Ethernet. We leave it blank.
8. The other item of interest on this window is the Server message queue, which allows you to have the install create a special message queue for logging Windows event logs. This can be desirable because password information can be logged in the events. Furthermore, separating the Windows event logs from the server joblog will make it easier to analyze the respective logs for messages related the Integrated Server Support or Windows-related issues.

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Virtual Ethernet port:
  Port . . . . . *vrteth1      *NONE, *VRTETH0, *VRTETH1...
  Windows internet address . . . 192.168.40.2
  Windows subnet mask . . . . . 255.255.255.0
  Associated port . . . . .      Name, *NONE
      + for more values
TCP/IP local domain name . . . . *SYS

TCP/IP name server system . . . . *SYS
      + for more values
Server message queue . . . . . *JOBLOG      Name, *JOBLOG, *NONE
  Library . . . . .              Name, *LIBL, *CURLIB
Event log . . . . .              *ALL        *ALL, *NONE, *SYS, *SEC, *APP
      + for more values

More...

```

Figure 4-43 Window four of the INSWNTSVR command

9. Page down (Figure 4-44 on page 87) and you are prompted to specify the size of the *system drive* and *install source drive*. It is acceptable to do *CALC for the *install source*, but you need to specify a size for the *system drive* that will be sufficient to accommodate the operating system, any applications that might utilize space, and consider that the Pagefile.sys and the registry are on this drive. If you are new to the product, you should note that this drive is not where you want user data to go so it does not need to be enormous. Also, note that you can create these disks in an Independent ASP (IASP) or a user-defined ASP. You will create additional virtual disks for your data after the install. The bottom portion of the window is omitted, because these are parameters that can be entered from the Windows GUI.


```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Server storage space sizes:
  Install source size . . . . . *CALC          500-2047, *CALC
  System size . . . . . 6000          1024-1024000, *CALC
Storage space ASP:
  Install source ASP . . . . . 1          1-255
  System ASP . . . . . 1          1-255
Server storage ASP device:
  Install source ASP device . . . . . Name
  System ASP device . . . . . Name
Convert to NTFS . . . . . *YES          *YES, *NO

```

Figure 4-44 Window five of INSWNTSVR

10. Page down again (Figure 4-45) and you can default this whole window, most of which can be entered in the Windows GUI. At the bottom are some parameters of note. The *shutdown timer* is how long the vary off command will wait for the Microsoft operating system to shut down before forcing the server down. This can be an important parameter if the server is running applications that require a considerable amount of time to shut down. There is also an *activation timer*, which might need to be adjusted if the remote system takes a long time to come up. The communications message queue will not be used much at this time so there is probably not a lot of reason to have it created. We will default all of these.

```

Shutdown timeout . . . . . 15          2-45
Activation timer . . . . . 120         30-1800
Communications message queue . . *SYSOPR   Name, *SYSOPR
Library . . . . .           Name, *LIBL, *CURLIB

```

Figure 4-45 Window six of INSWNTSVR

11. Page down to Figure 4-46 on page 88 and come to very significant parameters for the iSCSI implementation. The *storage path* and the *Virtual Ethernet path* signify the specific Host bus adapter that each function will use. If you specified to install an additional Virtual Ethernet (as we did), you need to put a plus in the prompt for additional values and specify the storage path for the *VRTETH1 port. The paths for Virtual Ethernet and storage do NOT have to be the same if you have more than one HBA.
12. See the following chapter, Chapter 5, "Implementing IBM Director Server" on page 107, for various scenarios, which utilize these functions. At the bottom of this window is where we are entering the names of precreated NWSCFG objects. These are the last parameters required.
13. If you do not choose to precreate the NWSCFG, there are three more windows and a lot of parameters to fill out.
14. However, because we are pre-creating the objects, we still need to press F9 to get to a parameter which has a default that will cause the command to stop. This parameter is IP Security Rule and it needs to be set to *none. See Figure 4-47 on page 88.

```
Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Storage path:
  Network server host adapter . > NHLIN07      Name
Virtual Ethernet path:
  Port . . . . . *VRTETHPTP      *VRTETHPTP, *VRTETH0...
  Network server host adapter . > NHLIN07      Name

  Port . . . . . > *VRTETH1      *VRTETHPTP, *VRTETH0...
  Network server host adapter . > NHLIN07      Name
    + for more values
Shutdown TCP port . . . . . 8700      1024-65535
Virtual Ethernet control port . 8800      1024-65535
Remote system NWSCFG . . . . . > RMKPWX545    Name, *DFT
Service processor NWSCFG . . . . . > SPKPKY358  Name, *DFT
Connection security NWSCFG . . . . . > CNNSEC1  Name, *DFT
Default IP security rule . . . . . *NONE
IP security rule . . . . . > *NONE      1-16, *DFTSECRULE, *NONE
```

Figure 4-46 Window seven of INSWNTSVR after adding storage path for additional VE port

```
IP security rule . . . . . *none      1-16, *DFTSECRULE, *NONE
```

Figure 4-47 Press F9 and change the default from *dftsecrule to *none

15. Press Enter and you should see INSWNTSVR is a long running command and it should continue to post progress messages while it creates the objects. After a period of time, it tells you to continue the install from Windows and the install should start the GUI install on the Windows console. See Windows install and post-install tasks.

Installing without precreating the objects

If you choose to have the install command create the NWSCFG objects, there are several other windows to configure. In this section, we run the install command and precreate the objects. We also fill in the parameters we defaulted previously.

- 1. Enter the command INSWNTSVR and press Enter. On Figure 4-48, input the name for the NWSD and press Enter.
- 2. Fill in the installation type, resource name, and Windows version. If you are not using the optical drive as we are not in this example, then specify the path to the install source.

```
Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Network server description . . . > INS      Name
Installation type . . . . . *FULL, *BASIC
Resource name . . . . . Name, *ISCSI
Windows server version . . . . . *WIN2000, *WIN2003
Windows source directory . . . . /cdimages/ws2003
```

Figure 4-48 Initial parameters for INSWNTSVR

3. Press enter to get additional prompts. The install option is the only option supported by iSCSI at this time.

Note: The parameters for TCP/IP port configuration only apply to LAN adapters that the System i5 can control, and thus these parameters are not applicable to iSCSI (Figure 4-49).

Install option	*INSTALL	*INSTALL, *UPGRADE
TCP/IP port configuration:		
Port	*NONE	*NONE, 1, 2, 3, 4
Windows internet address . . .		
Windows subnet mask		
Windows gateway address . . .		
+ for more values		

Figure 4-49 TCP/IP port configuration does not apply to iSCSI

4. Page down for more parameters. On Figure 4-50, we configured another Virtual Ethernet connection to be utilized for inter-LPAR communications. The associated port links this configuration with a Virtual Ethernet resource that would have been created from the hardware management console (HMC). This would be a resource that would have a hardware resource of 268C. That resource would need to be entered into the associated port field as illustrated. Also on this panel, we requested that a specific message queue be created to contain the Windows event log messages rather than have them go to the joblog for the server. This is desirable to be able to separate System i5 messages from Windows messages and to secure it from unauthorized access.

Install Windows Server (INSWNTSVR)		
Type choices, press Enter.		
Virtual Ethernet port:		
Port	*vrteth1	*NONE, *VRTETH0, *VRTETH1...
Windows internet address . . .	192.168.230.9	
Windows subnet mask	255.255.255.0	
Associated port	cmn07	Name, *NONE
+ for more values		
TCP/IP local domain name	*SYS	
TCP/IP name server system . . .	*SYS	
+ for more values		
Server message queue	win3k	Name, *JOBLOG, *NONE
Library		Name, *LIBL, *CURLIB
Event log	*ALL	*ALL, *NONE, *SYS, *SEC, *APP
+ for more values		

Figure 4-50 Virtual Ethernet configured for inter-LPAR communications

5. Page down again to Figure 4-51 on page 90. Configure the *install source drive* and *system drive* sizes. The *install source drive* is set to be calculated by the System i5. Realistically, the *system drive* that we have configured is large enough to contain the newly installed operating system, but is more than likely too small for many installations because the *registry* and the *Pagefile.sys* are on this drive. On the same panel, we configured the

Windows domain and organization prompts. These could have been defaulted as in the earlier example and filled in on the GUI portion of the install.

Important: Remember the Pagefile.sys is the swapper file. It normally is 1 and 1/2 times the installed memory. It can be adjusted, but if it is too small, performance suffers. If it is not large enough, you might not be able to capture memory dumps if that occasion should arise. Also, it can be moved to a separate drive, but this precludes capturing a memory dump at all. Many customers forget to account for this when sizing the drive.

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Server storage space sizes:
  Install source size . . . . . *CALC      500-2047, *CALC
  System size . . . . . > 5000      1024-1024000, *CALC
Storage space ASP:
  Install source ASP . . . . . 1      1-255
  System ASP . . . . . 1      1-255
Server storage ASP device:
  Install source ASP device . . . . . Name
  System ASP device . . . . . Name
Convert to NTFS . . . . . *YES      *YES, *NO
To workgroup . . . . .
To domain . . . . . > Windom2
Full Name . . . . . > itso

Organization . . . . . > ibm

More...

```

Figure 4-51 Allocating system drive and install source drives

6. Page down to Figure 4-52 on page 91 and additional Windows prompts are presented. At the top of the window, choose whether to Synchronize date and time with the System i5. Most customers set this, but if the network is being used to serve time, you want to turn this off.
7. The Propagate domain user specifies if this server will be used to enroll System i5 users in the Windows domain.
8. The Windows parameters, that is, license key and license mode, could have been deferred to the Windows portion of the install. Other parameters of interest are in bold at the bottom.
9. The *shutdown timer* sets the amount of time the VARYCFG command waits to force the NWSD down while waiting for the Windows operating system to shut down. This can be very important depending on what applications are running on the server. The *activation timer* might also be important if the server is taking a long time to power on. The last parameter is the communications queue and this is not used much initially so it is probably not worth setting, but it is available if you want to segregate the messages from QSYSOPR.

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Language version . . . . . *PRIMARY      *PRIMARY, 2911, 2922, 2923...
Synchronize date and time . . . *YES      *YES, *NO
Propagate domain user . . . . *YES      *YES, *NO
Windows license key . . . . . dmrhm-8qpqh-x76nx-jhrrb-98ujh

License mode:
  License type . . . . . *PERSEAT      *PERSEAT, *PERSERVER
  Client licenses . . . . . *NONE      5-9999, *NONE
  Terminal services . . . . . *NONE      *NONE, *TSENABLE...
  Restricted device resources . . *NONE      Name, *NONE, *ALL...
      + for more values
Shutdown timeout . . . . . 15      2-45
Activation timer . . . . . 120     30-1800
Communications message queue . *SYSOPR    Name, *SYSOPR
  Library . . . . .      Name, *LIBL, *CURLIB

More...

```

Figure 4-52 Choose to synchronize time and whether to enroll users

10. Page down yet again to Figure 4-53. This window is important if you decide to implement the environments discussed in the subsequent chapters. Specify which HBA to use for the iSCSI (storage requests) and which to use for the Virtual Ethernet. They can be the same HBA or different HBAs. We default the NWSCFG prompts, because we use the command to create them. This results in files which have defaulted names such as the following example illustrates in Figure 4-53.

```

Work with NWS Configuration

Type options below, then press Enter.
1=Create 2=Change 4=Delete 5=Display 6=Print 8=Work with status

System:

Opt  Name      Type      Text
   IAS300RS    *RMTSYS   I Team Blade Server
   ITS03CN     *CNNSEC
   ITS03RM     *RMTSYS
   ITS03SP     *SRVPRC

```

Figure 4-53 Default names of the NWS

Important: These names cannot be changed by using the iSeries Navigator by using the commands CHGNWSCFG or WRKNWSCFG.

```
Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Storage path:
  Network server host adapter . > NHLIN07      Name
Virtual Ethernet path:
  Port . . . . . *VRTETHPTP      *VRTETHPTP, *VRTETH0...
  Network server host adapter . > NHLIN07      Name

  Port . . . . . > *VRTETH1      *VRTETHPTP, *VRTETH0...
  Network server host adapter . > NHLIN07      Name
    + for more values
Shutdown TCP port . . . . . 8700      1024-65535
Virtual Ethernet control port . 8800      1024-65535
Remote system NWSCFG . . . . . *DFT      Name, *DFT
Service processor NWSCFG . . . . *DFT      Name, *DFT
Connection security NWSCFG . . . *DFT      Name, *DFT
```

Figure 4-54 Storage and VE paths and default the NWSCFG

11. Press Enter (Figure 4-55) and get one more parameter appended to this window. It must be defaulted.

```
Default IP security rule . . . . *NONE
```

Figure 4-55 Additional parameter not used at this time

12. Press Enter (Figure 4-56) and get another single prompt appended.

```
Initialize service processor . . *NONE      *MANUAL, *SYNC, *AUTO, *NON
```

Figure 4-56 Additional parameter: *AUTO is not valid at this time

- 13. Page down and change Enable unicast to *yes, and enter the serial number of the service processor in Figure 4-57 on page 93.
- 14. Press Enter and another prompt is appended. You can default the Service Processor Name.
- 15. Press Enter again and the parameters are appended that define the NWSCFG objects. The user name and password would be USERID/PASSWORD if you have not changed the defaults. This is the profile that allows access to the service processor. The serial number and manufacturer information are specific to the server hardware. On an xSeries, it would be the tag on the box. For a BladeCenter, this is the information relative to the specific blade.

```

Enable unicast . . . . . > *YES          *NO, *YES
Enclosure identifier:
  Serial number . . . . . > KPKY358      Character value, *AUTO
  Manufacturer type and model . > 86772XX  Character value
Service processor name . . . . . *SPINTNETA
SP internet address . . . . . > '109.75.92.241'
SP authentication:
  User name . . . . . > ITCDEM02
  User password . . . . . > test
SP certificate identifier:
  Component . . . . . *NONE             *NONE, *COMMONNAME, *EMAIL...
  Compare value . . . . .
Remote system identifier:
  Serial number . . . . . > KPDL842      Character value, *EID
  Manufacturer type and model . > 884301U  Character value

```

Figure 4-57 Finishing service processor configuration, starting remote system configuration

16. Page down to Figure 4-58 and get more prompts for NWSCFG configuration. We have preconfigured the Blade with a CHAP secret and CHAP name, so enter that here. The rest of this window can be defaulted because we are doing DHCP for the blade.

17. Page down.

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

CHAP authentication:
  CHAP name . . . . . chap
  CHAP secret . . . . . chap
Boot device ID:
  Bus . . . . . *SINGLE      0-255, *SINGLE
  Device . . . . .          0-31
  Function . . . . .          0-7
Dynamic boot options:
  Vendor ID . . . . . *DFT   Character value, *DFT
  Alternate client ID . . . . *ADPT Character value, *ADPT

```

Figure 4-58 Specifying CHAP name and CHAP secret

18. Page down to Figure 4-59 on page 94 and configure the specific remote systems iSCSI and Lan connections. Note that if you had multiple iSCSI interfaces, you would need to go back and add it. Because this is a Blade configuration, there is only one more possible port. If it was an xSeries, there could be three more.

```

Install Windows Server (INSWNTSVR)

Type choices, press Enter.

Remote interfaces:
  SCSI interface:
    Adapter address . . . . . > 00C0DD07597A Hexadecimal value
    Internet address . . . . . > '128.168.207.1'
    Subnet mask . . . . . > '255.255.0.0'
    Gateway address . . . . . > '128.168.200.1'
    iSCSI qualified name . . . . . *GEN
  LAN interface:
    Adapter address . . . . . > 00C0DD075979 Hexadecimal value
    Internet address . . . . . > '128.168.208.1'
    Subnet mask . . . . . > '255.255.0.0'
    Gateway address . . . . . > '128.168.200.1'
    Text 'description' . . . . . *BLANK

```

Figure 4-59 Remote system configuration

19. Press Enter and if everything has been entered correctly, a message states that the command is a long running one and it starts to create storage spaces.

Note: If you have to retrieve the command to change some parameters, the password for the service processor authentication and the CHAP secret are blanked out.

This example was for a Blade, but you follow the same process for an xSeries server.

Tasks accomplished from the Microsoft operating system

In one of the installation examples documented here, we have chosen not to enter the license information, the domain or work group information, or any of parameters that can be entered from the Windows GUI. If you choose to use the default parameters, you have to complete some installation tasks and there are other tasks that you need to complete:

1. Accept the license agreement.
2. Enter the license key for the software.
3. Choose a domain or workgroup.

Additional tasks

There are several additional tasks to perform:

1. Install Ethernet LAN drivers for the external Ethernet and configure them:
 - a. The Integrated Server Support installs services and drivers to interface with the System i5. It is not able to install the drivers for the xSeries or Blade hardware. After the install, there will be no drivers installed for the external Ethernet connection.
 - b. To verify the network connections that are installed:
 - i. Click **START** → **Expand All Programs** → **Control Panel** → **Network connections** to display the network connections that installed. In Figure 4-60 on page 95, there are three displayed because we installed an additional Virtual Ethernet adapter, so we have a connection for the QLogic (iSCSI adapter), the Virtual Ethernet point-to-point, and the Virtual Ethernet that we created additionally.

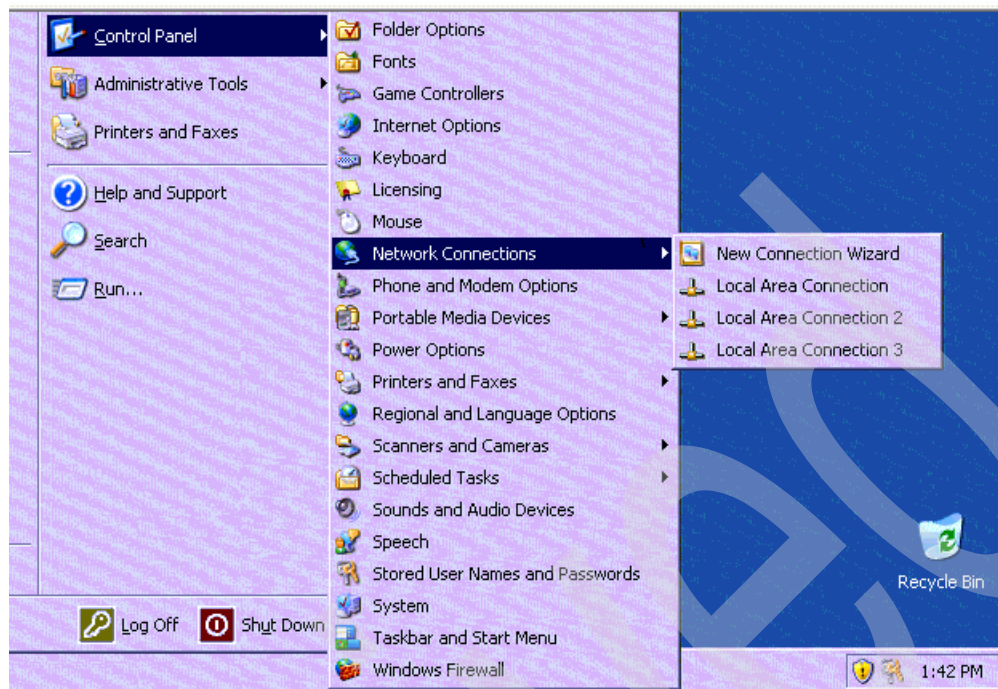


Figure 4-60 Verifying which LAN connections were installed

- ii. Right-click each of the connections and click **properties** to identify the connection. In Figure 4-61, we show the QLogic adapter.

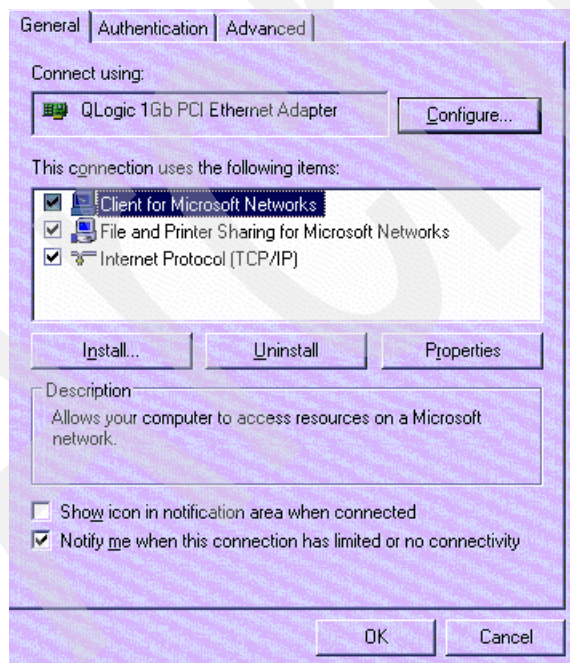


Figure 4-61 Identifying LAN connections

- iii. To verify that the Ethernet hardware is seen (Figure 4-62 on page 96), click **Start** and select **Computer Management**.

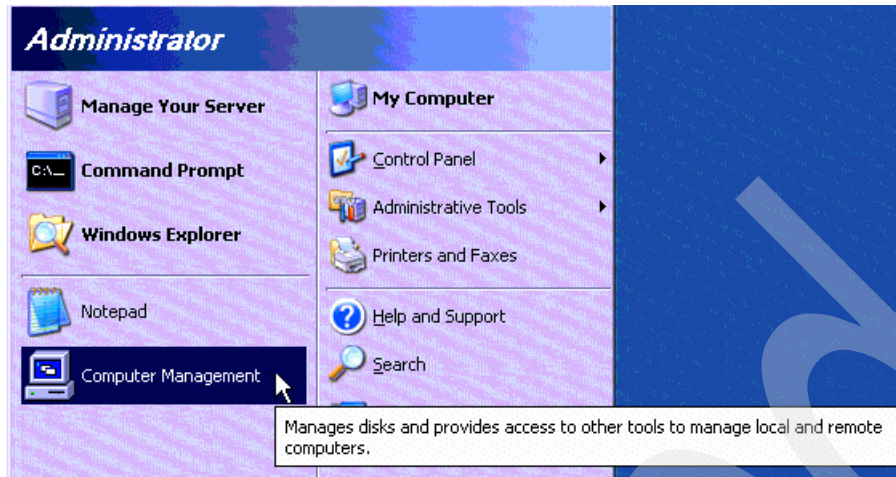


Figure 4-62 Select Computer Management to verify Ethernet hardware

- iv. Double-click **Computer Management** and click **Device manager** (Figure 4-63).

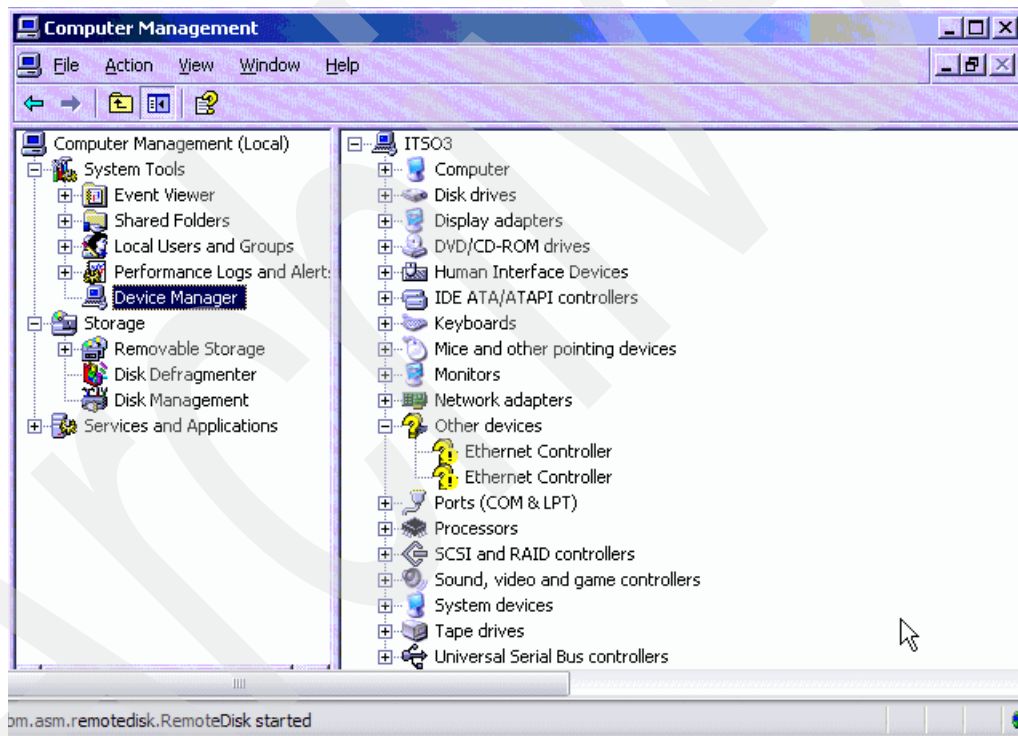


Figure 4-63 Ethernet hardware is seen but it has a problem

- v. Double-click one of the Ethernet controllers and the status tells you that the driver is not installed (Figure 4-64 on page 97).

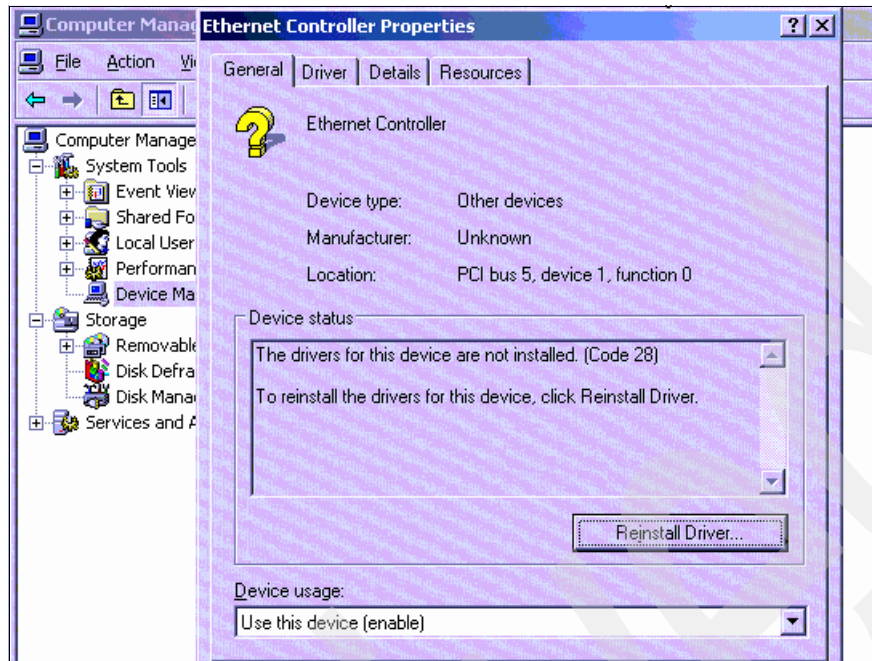


Figure 4-64 Confirming drivers do not install

- vi. Locate the applicable driver by going to the supported models page and clicking the link to download the firmware next to the xSeries or Blade. In Figure 4-65, we are selecting the blade. Note we erased the URL because it was an internal site. The correct URL would be:

<http://www.ibm.com/servers/eserver/iseries/integratedxseries/iscsiservermodels/>

x346 server configurations				1 (Download firmware)
8840-45Y	884045U	1-2	3.6 GHz	2
8840-05Y	884005U	1-2	2.8 GHz	2
8840-55Y	884055U	1-2	3.8 GHz	2
x236 server configurations				1 (Download firmware)
8841-45Y	884145U	1-2	3.6 GHz	2
8841-05Y	884105U	1-2	2.8 GHz	2
8841-55Y	884155U	1-2	3.8 GHz	2
BladeCenter blade models supported with iSCSI				
Model Number	US Part Number	Number of Processors	Speed of Processors	Notes
HS20 blade configurations				1 (Download firmware)
8843-01U	884301U	1-2	2.8 GHz	2
8843-E9U	8843E9U	2	3.8 GHz	2
BladeCenter chassis models supported with iSCSI				
Model Number	US Part Number	Number of Blades	Notes	
Chassis configurations				1 (Download firmware)
8677-3XY	86773XU	1-14		
8677-3EY	86773EU	1-14		

Figure 4-65 Select Download firmware link next to the model of server or blade

- vii. This takes you to the drivers download page for your model (Figure 4-66 on page 98). Scroll down until you see the Networking section and download the driver for the adapter.

IBM Hard Disk Drive Update Program (Windows package)	03 Mar 2006 v1.18	<--	<--
Networking	BladeCenter HS20 (8678)	BladeCenter HS20 (8832)	BladeCenter HS20 (8843)
Broadcom Firmware Update Utility	25 Jan 2006 v1.20.15	<--	<--
Broadcom NetXtreme Gigabit Ethernet Software CD <ul style="list-style-type: none"> • Microsoft Windows NT4 • Microsoft Windows 2000 • Microsoft Windows Server 2003 • Netware • Linux • SCO OpenServer • Unixware 	05 Dec 2005 v8.3.6	<--	<--
Broadcom NetXtreme Gigabit Ethernet driver for Microsoft Windows XP and Windows Server 2003 32-bit digitally signed (from CD 7.0.2) Windows update package	19 May 2004 v7.33	19 May 2004 v7.33	19 May 2004 v7.33
NetXtreme Gigabit Ethernet Software CD digitally signed (except 64 bit driver)	n/a	03 Jul 2003 v6.7.7	n/a

Figure 4-66 Scroll down to Networking and download the driver for your server

- viii. Follow the instructions to expand the executable and copy the files to diskette.
- ix. Put the diskette in the diskette drive for the machine.
- x. Return to the Device Manager on the console and right-click the Ethernet controller with the yellow question mark (Figure 4-67).

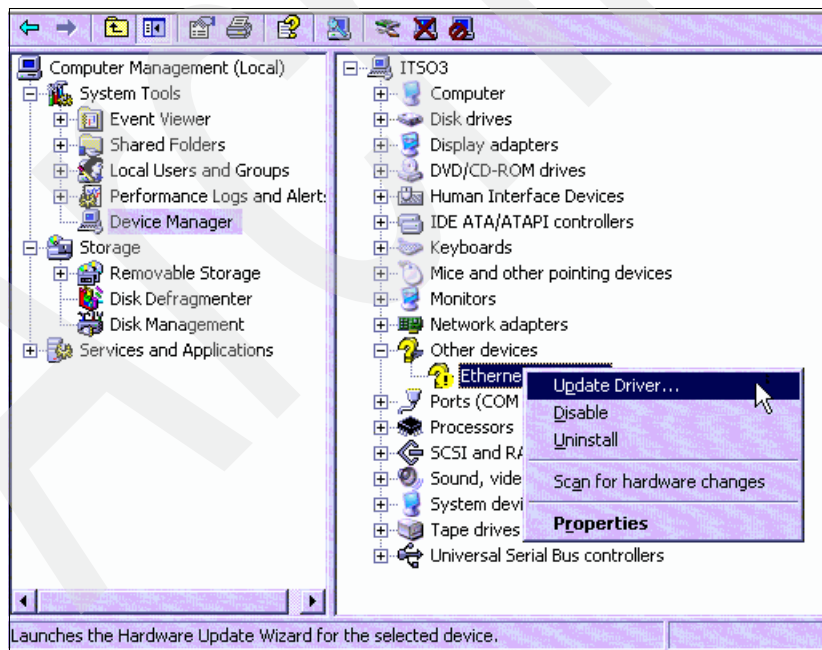


Figure 4-67 Update the Ethernet driver

- xi. This produces the update wizard which updates the driver.
- xii. When finished, the yellow question mark disappears and you can configure a connection added to the network connections.

2. Create additional virtual disks for user data and link them. To create and link new storage spaces in iSeries Navigator (Figure 4-68):
 - a. Expand **Integrated Server Support**.
 - b. Right-click **All Virtual Disks**.

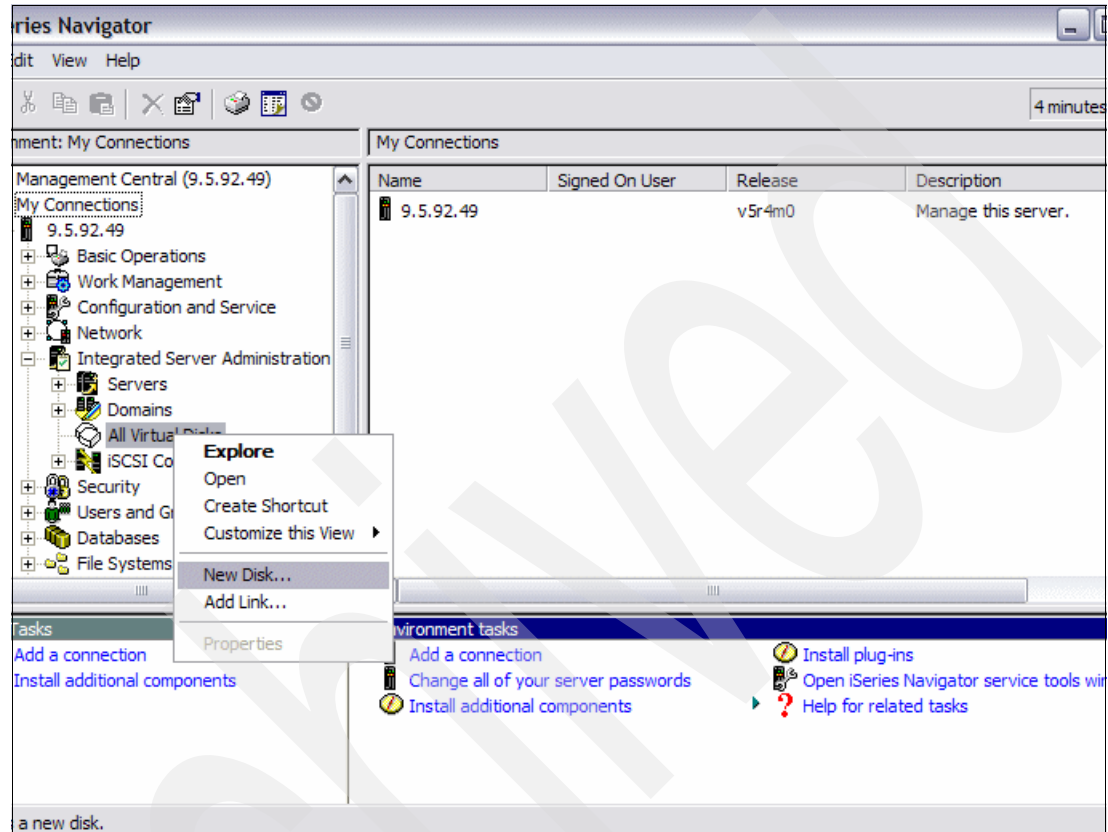


Figure 4-68 Creating new virtual disk

- c. In Figure 4-69 on page 100, specify a drive name (consider using a naming convention to make referencing specific disks easier. We named it after the NWSD and added a 3 to denote it is the first disk after ITSO31 and ITSO32, which are the drives created by the install.
- d. Select the disk pool. We took the default.
- e. Specify the file system which we left as NTFS.
- f. Specify to link disk to the server and select the server.
- g. Dynamic link is the only valid link with iSCSI.
- h. Specify the storage path, that is, the NWSH it will use.
- i. Specify the Access, normally Exclusive - Update.
- j. Click **OK** and the message returns that it is linked and needs to be formatted.

New Disk - 9.5.92.49

Disk drive name:

Description:

☐ Initialize disk with data from another disk

Source disk:

Capacity: ☐ GB ☒ MB

Disk pool:

Planned file system:

Windows cluster quorum resource attributes

Cluster domain name:

Cluster connection configuration:

Virtual Ethernet:

Internet address:

Subnet mask:

☒ Link disk to a server

Link to server:

Link type:

Link sequence position:

Storage path:

Access to disk drive:

Figure 4-69 New Disk description

- k. Configure new disks in Windows (Figure 4-70 on page 101):
 - i. Expand **Administrative tools**.
 - ii. Select **Computer Management**.

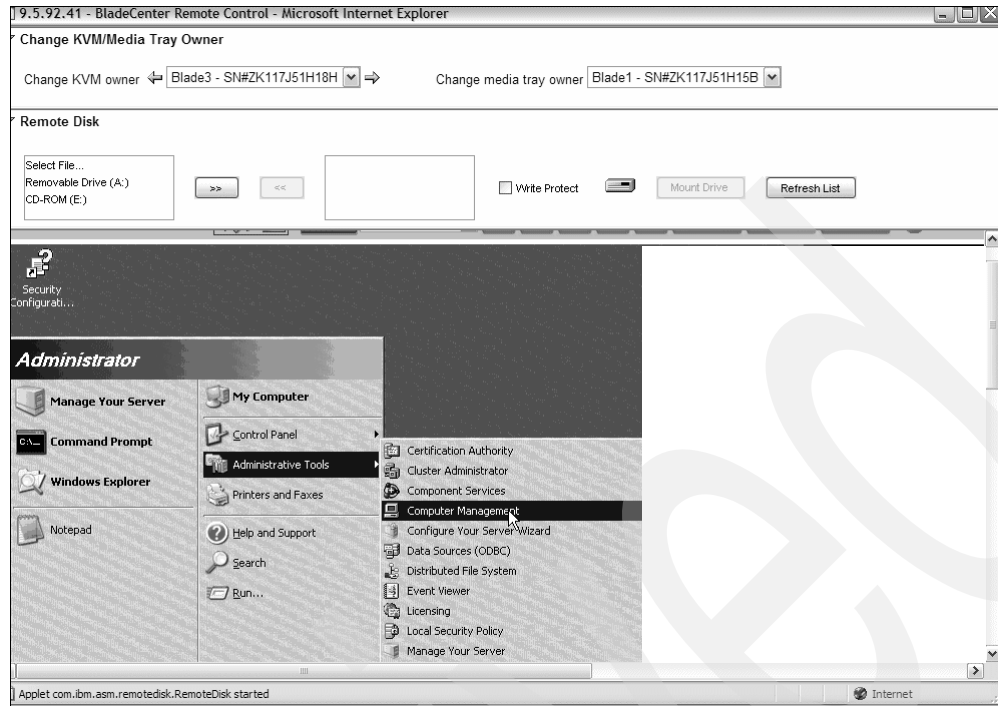


Figure 4-70 Access Computer Management to configure disks

iii. Click **Disk Management**.

iv. Look on the bottom panel for additional disk that shows as unallocated (Figure 4-71).

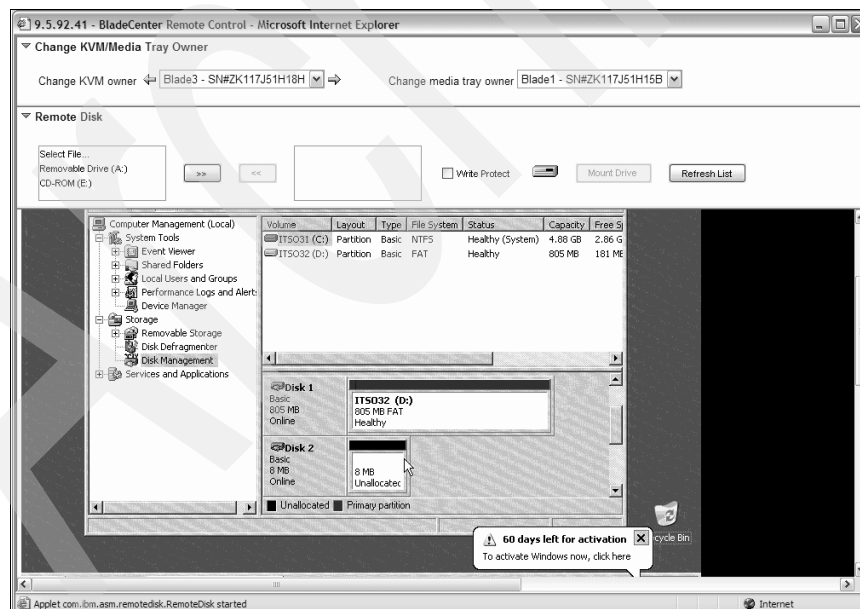


Figure 4-71 From Remote Control, viewing new disk on Blade server

v. You can choose whether to convert the disk to a dynamic disk or make it a basic disk. If you plan to add it to a volume set, it must be a dynamic disk. Dynamic disks can have some issues, so you might want to look at the documentation regarding dynamic disks before deciding to do this. To convert to dynamic disk, right-click the

right side of the panel labelled Disk 2 in Figure 4-72 and take the option to **Convert to Dynamic Disk**. To make it a basic disk, right-click in the panel labelled unallocated as shown in Figure 4-72 and select **New Partition**.

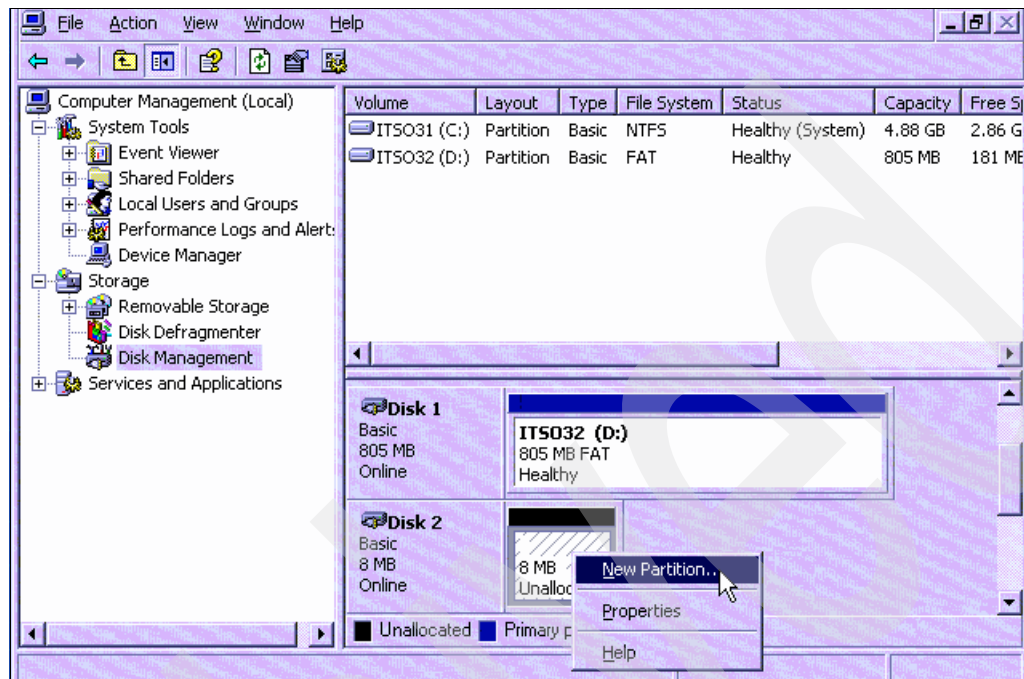


Figure 4-72 Select new partition to create a basic disk

- vi. We make a Basic disk with one partition. Selecting the New Partition starts a wizard (Figure 4-73).

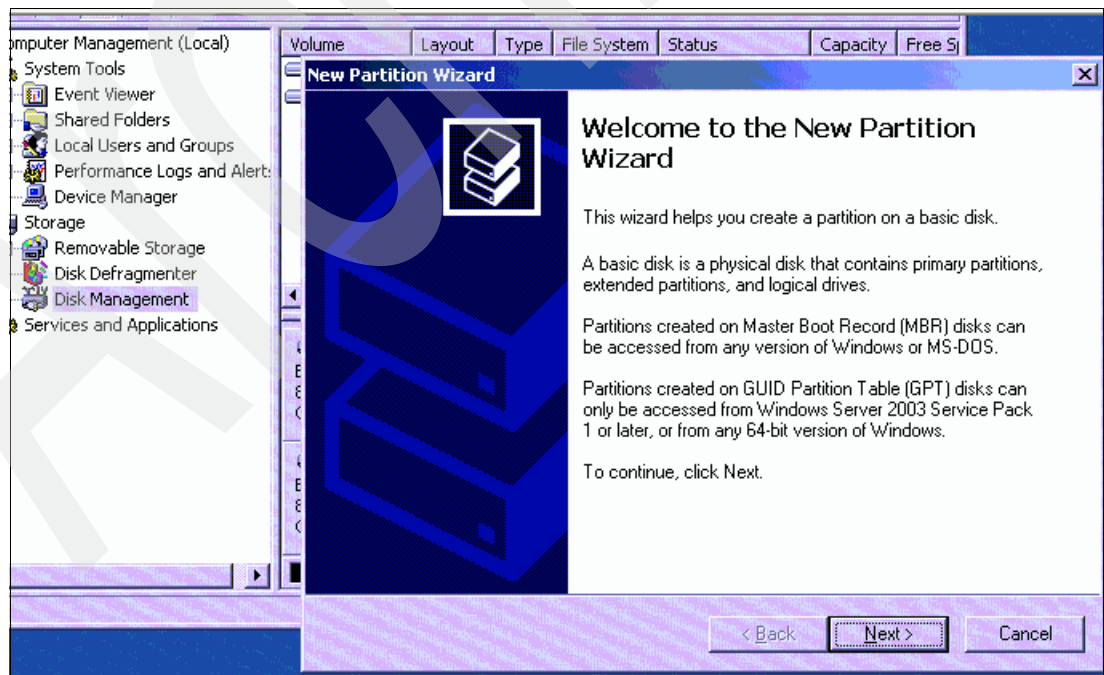


Figure 4-73 New partition wizard to create a basic disk

- vii. Choose to create a primary partition.

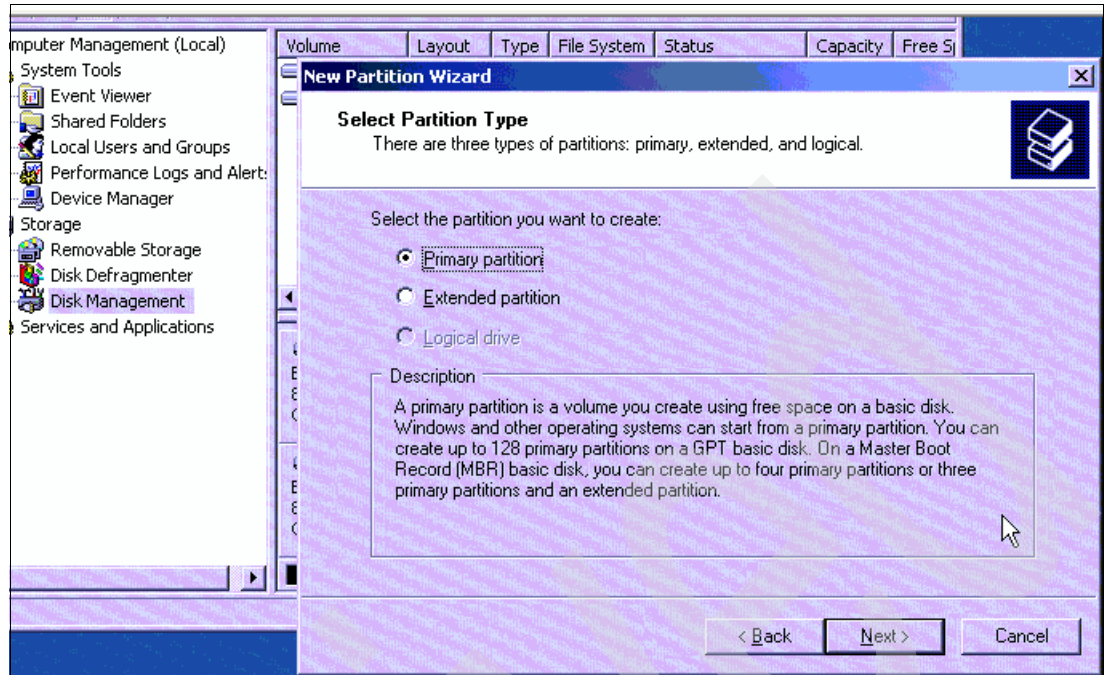


Figure 4-74 Creating a primary partition to create a basic disk

viii. Click **Next** and specify that partition size is the size of the whole amount of allocated space that is to make a single partition (Figure 4-75). This important.

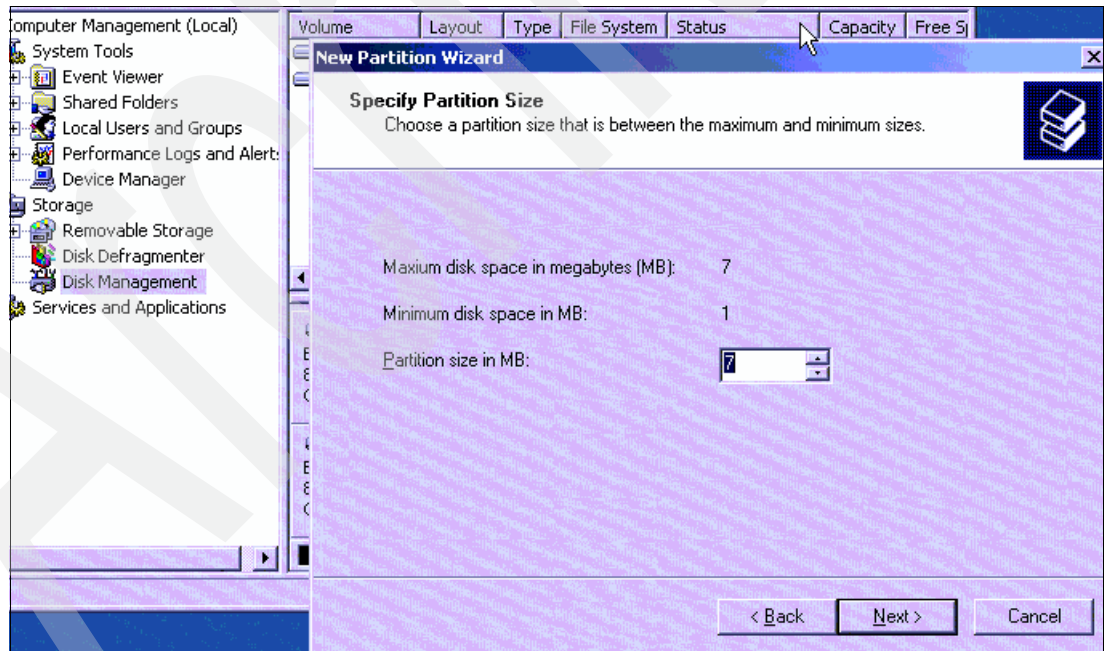


Figure 4-75 Make the disk a single partition by allocating all of the space to the partition

Important: We highly recommend that in order to optimize the performance of iSCSI drives, you observe the following recommendations:

- ▶ One disk partition for virtual drive
- ▶ One gigabyte or larger virtual disks
- ▶ NTFS file system formatted with 4096 or larger cluster sizes

The virtual disks are optimized for this environment. To add a partition, add another virtual disk and add it to a spanned volume if desired. See the following URL for more information:

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/index.jsp?topic=rzahq/rzahqiscsiattachedperf.htm>

ix. Click **Next** and assign a drive letter (Figure 4-76).

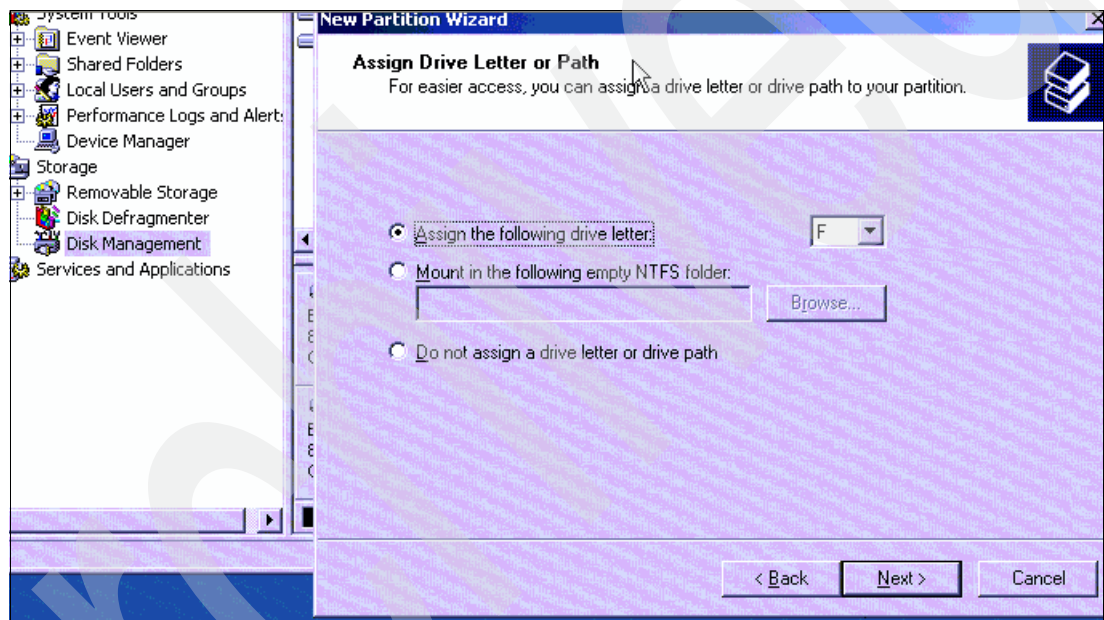


Figure 4-76 Assign a drive letter

x. In Figure 4-77 on page 105, format the drive as NTFS and set the allocation size to 4096 or larger.

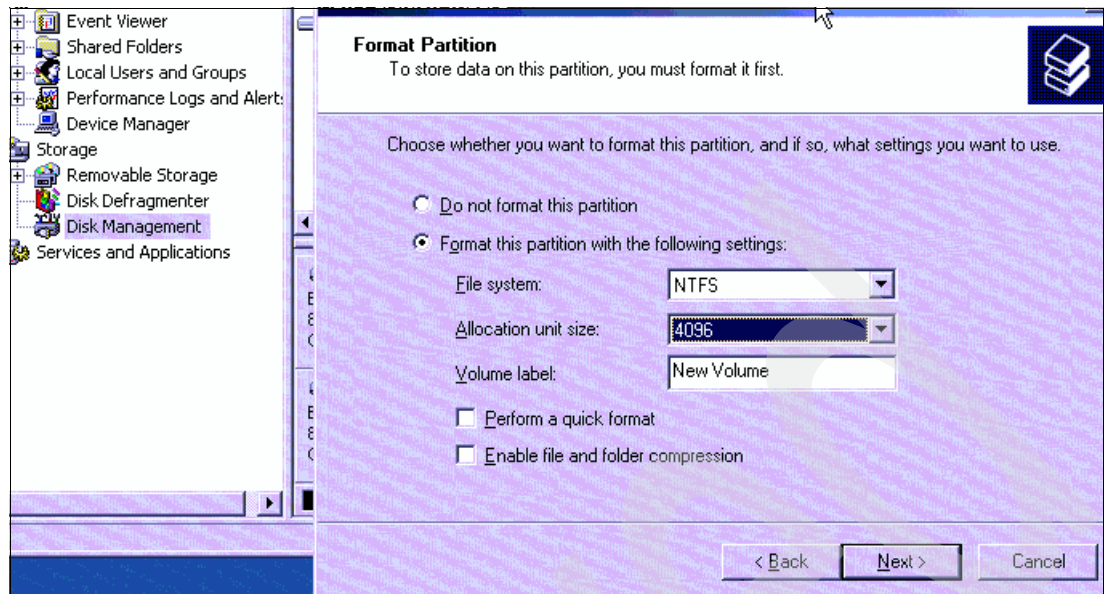


Figure 4-77 Formatting the drive

xi. Click **Next**. In Figure 4-78, review the information, and click Finish if it is correct.

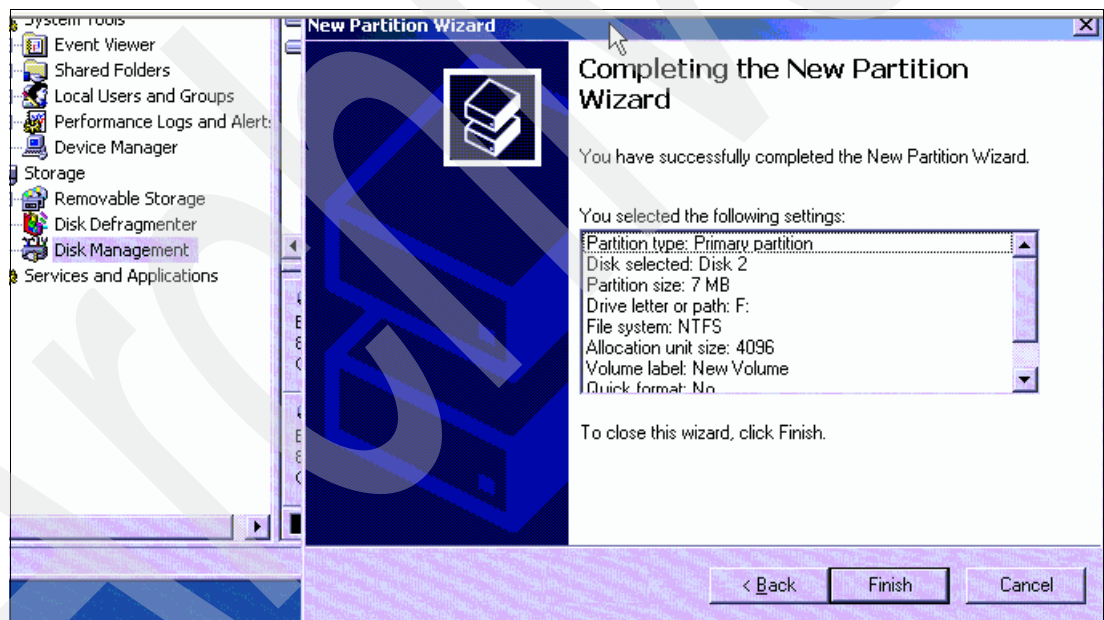


Figure 4-78 Completing the formatting of the disk

xii. Verify the status of disk (Figure 4-79 on page 106).

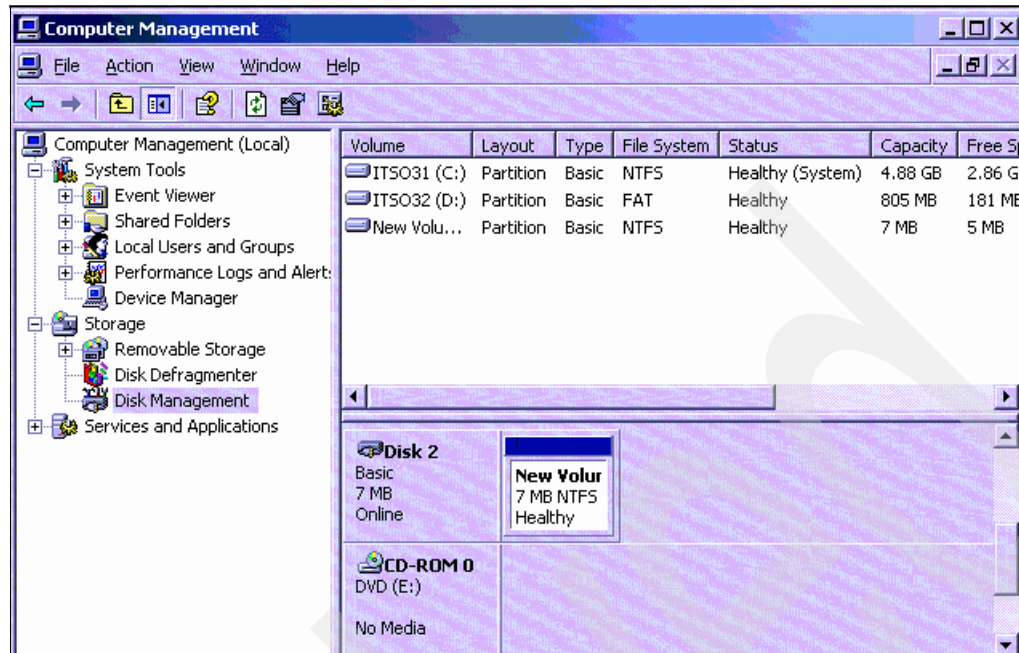


Figure 4-79 Verifying status of the new disk

The new disk is now configured.



Implementing IBM Director Server

This chapter describes the installation of IBM Director Server in an iSCSI environment. It describes the software and hardware requirements as well as a fast path to get IBM Director Server installed on the iSeries system connected to an iSCSI integrated server.

5.1 Introduction to IBM Virtualization Engine

The Virtualization Engine can help you automate the management of the resources based on your business goals and make basic systems management of multiple systems possible.

The IBM Virtualization Engine is a set of technologies and systems services that can help you aggregate pools of resources and get a consolidated view of them throughout your IT environment. It uses key IBM virtualization technologies to give you a logical rather than physical view of data, computing power, storage capacity, and other resources.

There are two types of the Virtualization Engine, both with their own tools (components).

The Virtualization Engine Systems Edition provides the following advantages:

- ▶ Pools all computing resources (servers, storage, and network appliances, both IBM and others) into one virtual environment
- ▶ Simplifies infrastructure and reduces management complexity by building integrated virtualization platform technologies into every box
- ▶ Provides consistency across the IBM and System Storage™ brands
- ▶ Uses a common open interface
- ▶ Provides an integrated approach to enterprise-wide virtualization

The Virtualization Engine Management Collection provides multiplatform system tools, such as:

- ▶ IBM Virtualization Engine console
- ▶ IBM Enterprise Workload Manager
- ▶ IBM Director Multiplatform
- ▶ Resource Dependency Service

IBM Virtualization Engine products and their components can be installed separately. The IBM Virtualization Engine Base Support is mandatory and on top of the Base Support, you can install these components such as IBM Director Server. This is the same as iSeries Navigator, base support is mandatory and the components such as Basic Operations, Network, and so on are optional.

More in-depth information about IBM Virtualization Engine is available on the Web site:

<http://www-03.ibm.com/servers/eserver/about/virtualization/>

5.2 IBM Director Server in an iSCSI environment

IBM Director Server Version 5.1.0 of Virtualization Engine must be installed to get iSCSI integrated server to work. It is part of both types of Virtualization Engines (Systems Edition and Management Collection). There is no need to install all the tools of Virtualization Engine if only IBM Director Server is needed. We describe a simple fast path installation in Chapter 6, "Managing integrated iSCSI environments" on page 149, if IBM Director Server Version 5.1.0 (5722-DR1) is not installed already.

IBM Director is used for remote server discovery and management of iSCSI attached servers. No IBM Director interface such as IBM Director Console is needed in the iSCSI environment. Just installing and starting IBM Director is enough.

5.2.1 Why IBM Director Server

iSCSI integrated environment on iSeries uses IBM Director Server for:

- ▶ Remote server and service processor discovery, finding the server on the network.
- ▶ Power control turning the server on or performing an operating system shutdown for appropriate i5/OS vary configuration commands.
- ▶ Power status retrieval.
- ▶ Configuration of the remote server. Some remote server functions can be configured from the iSeries remotely through the remote server's service processor.

Note: Each partition having iSCSI Windows connections needs to have IBM Director Server installed. There is more information about this later in this chapter.

5.2.2 Start IBM Director Server

IBM Director Server depends on TCP/IP. TCP/IP must be started for IBM Director Server to function. Because IBM Director Server is an i5/OS TCP server, it can be configured to autostart when TCP is started. We recommend that the IBM Director TCP server is configured to automatically start. This ensures that IBM Director Server is available when the iSCSI Host Bus Adapter (HBA) for iSeries needs it.

To configure the IBM Director TCP server to automatically start when TCP/IP is started, use iSeries Navigator to perform the following steps (Figure 5-1 on page 110):

1. Expand **yourserver**.
2. Expand **Network** → **Servers**, and click **User-Defined**.
3. At the right pane, right-click **IBM DIRECTOR** and select Properties.
4. Select the **Start when TCP/IP is started** option. Click **OK**.

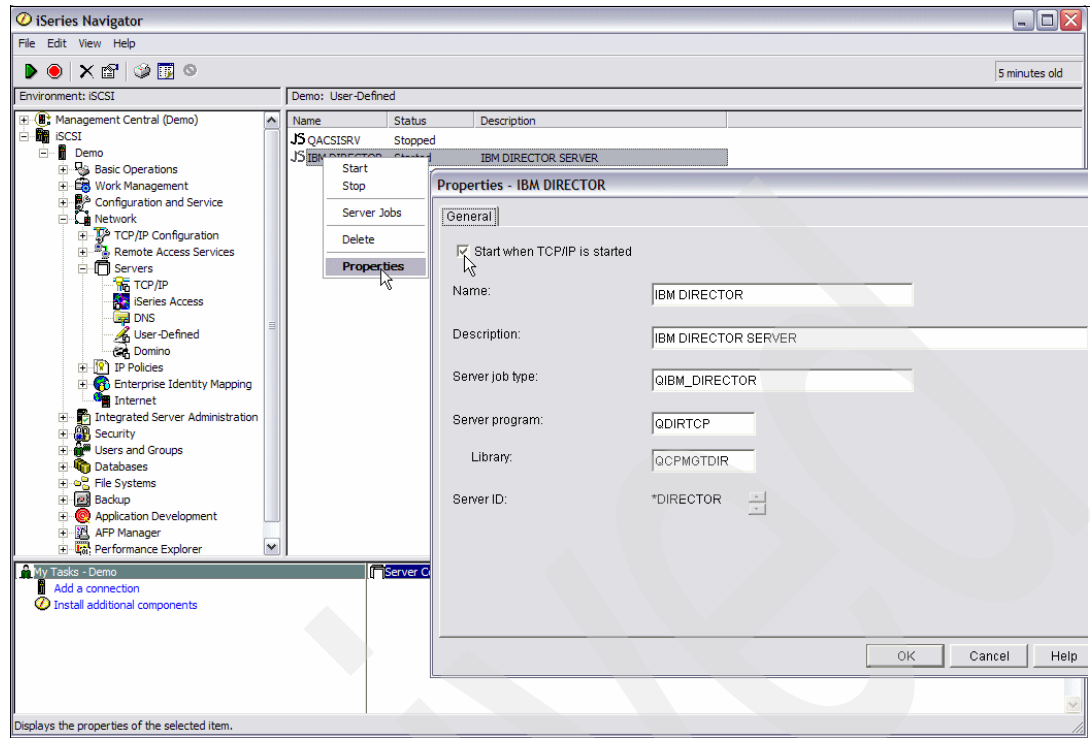


Figure 5-1 Properties of IBM Director Server

You can also use the Change TCP/IP Server (**CHGTCPSVR**) command.

1. On a 5250 emulation command line, enter **CHGTCPSVR** and press Enter or F4.
2. On the window titled Change TCP/IP server (Figure 5-2 on page 111), type for the parameter Server special value (SVRSPCVL) ***DIRECTOR**, and press Enter.

3. More parameters are shown in Figure 5-2, change the Autostart (AUTOSTART) parameter into *YES and press Enter.

Change TCP/IP Server (CHGTCPSVR)

Type choices, press Enter.

Server special value	> *DIRECTOR	Character value
Program to call	QDIRTCP	Name, *SAME
Library	QCPMGDIR	Name
Server name	'IBM DIRECTOR'	
Server type	'QIBM_DIRECTOR'	
Autostart	*YES	*SAME, *YES, *NO
Text 'description'	'IBM DIRECTOR SERVER'	

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 5-2 CHGTCPSVR CL command

If IBM Director Server is not automatically started, it will be started during the vary-on process of the NWSD, but it will take several minutes or more before the server is active. In Figure 5-3 on page 112, you can also use iSeries Navigator to start the IBM Director TCP server:

1. Select **Network** → **Servers** → **User-Defined**.
2. Right-click **IBM DIRECTOR** and select **Start**.

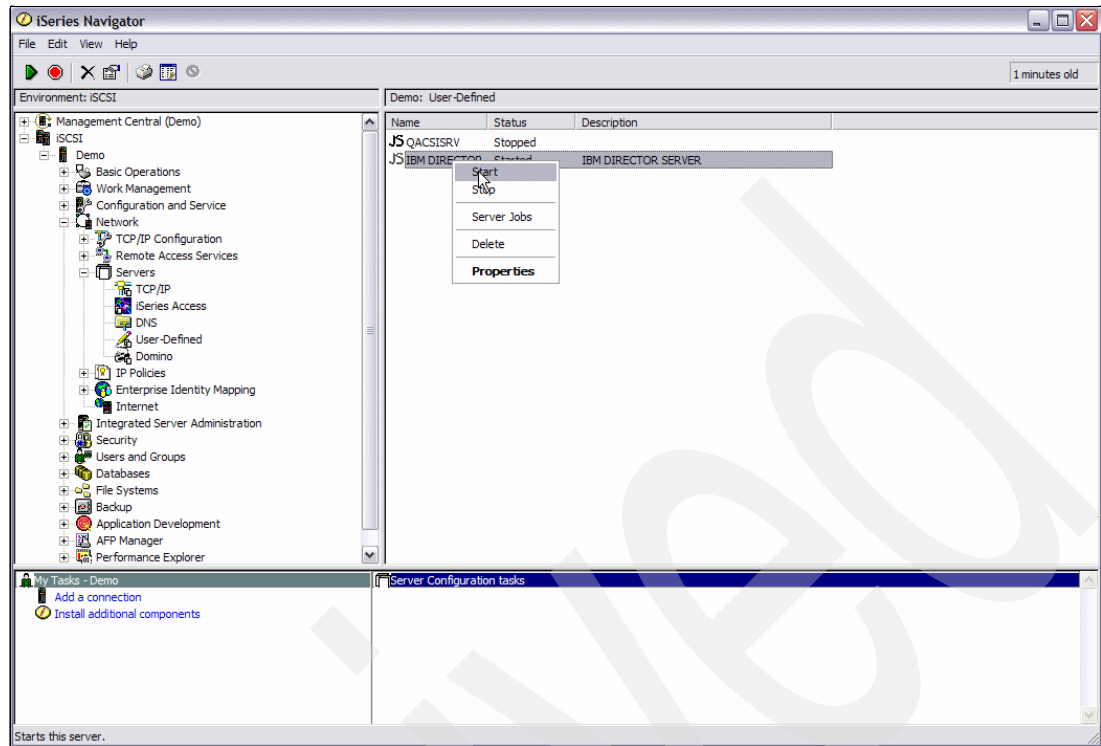


Figure 5-3 Start IBM Director Server

Note: It will take several minutes or more for the IBM Director server to start. You can view the status of the start process by refreshing the iSeries Navigator list until the IBM DIRECTOR server shows a status of Started. At this time it is not completely started. To make sure it is started, you can check the real status as follows:

1. Open a 5250 emulation session.
2. Execute the command **STRQSH**.
3. Type **cd /qibm/proddata/director/bin** and press Enter.
4. Type **twgstat** and press Enter. This will return the status of IBM Director Server. When this returns **Active**, IBM Director Server is completely started.

5.2.3 iSeries Hardware requirements

The minimum commercial processing workload (CPW), storage pool size, and disk space needed by the IBM Director components are in Table 5-1.

Table 5-1 iSeries Server hardware requirements

Requirements	IBM Director Server
Relative system performance	150 CPW
Storage pool size	500 MB
Disk Space	500 MB

5.2.4 iSeries software requirements

IBM Virtualization Engine (5733-VE2) is on the i5/OS CD, which you receive when you order i5/OS release 540. Ensure that the following products and options are installed on the iSeries (i5/OS V5R4) server on which you plan to install IBM Director, along with the latest cumulative PTF packages or Group PTFs. Also make sure to install the recommended fixes for IBM Director Server 510 by going to the Web site:

<http://www.ibm.com/pc/support/site.wss/document.do?ln docid=SERV-DIRECT>

The installation instructions for IBM Director server update is described in 5.6.1, “IBM Director server update” on page 139.

The products and options in Table 5-2 are required to successfully install and run IBM Director.

Table 5-2 iSeries Server Software product and options requirements

Products and options	Order Number
Extended Base Directory Support, Option 3	5722SS1
Java Developer Kit 1.4, Option 6	5722SS1
OS/400 - QSHELL, Option 30	5722SS1

Note: IBM Director is a no-cost option of the Virtualization Engine (5733-VE2) and has additional software requirements.

5.2.5 IBM Director Agent (5722-DA1) installed

It is most likely not the case that you have IBM Director Agent installed on your iSeries, because IBM Director Server includes Agent code. In the case that you do have IBM Director Agent installed in your environment on the iSeries system, follow these steps:

1. Check if the LPP 5722-DA1 is installed by using the command **DSPSFWRSC**. If it is installed, continue with step 2.
2. At an i5/OS command prompt on the system on which IBM Director Agent is installed, type the following command and press Enter:

```
DLTLICPGM LICPGM(5722DA1)
```

3. To remove IBM Director user data from the i5/OS management server, delete the /qibm/userdata/director/ directory using the following steps in QSHELL:
 - a. On a 5250 emulation command line, type **strqsh** and press Enter.
 - b. The QSH Command entry display opens, type the following on the command line and press Enter:

```
rm -rf /qibm/userdata/director/
```
 - c. On the display, the results of the command appears. When the dollar sign character (\$) returns, the command is finished.
4. When this is done, you can continue to install IBM Director Server in 5.3, “Fastpath install” on page 114.

5.2.6 IBM Director Server already installed or older version

If IBM Director Server is already installed as part of the installation of Virtualization Engine Version 1, you have to upgrade to Virtualization Engine Version 2, which includes IBM Director Server Version 5.10. You can install the new version of IBM Director Server over an existing install. The IBM Director Server part of the install will migrate IBM Director Server to the new version. To upgrade IBM Director, follow the steps for a new installation, including any planning and environment preparation that might be necessary. So you can use the fastpath installation in 5.3, “Fastpath install” on page 114.

If you are running one of the following versions of IBM Director on a supported operating system, you can upgrade to IBM Director 5.10:

- ▶ IBM Director 4.10
- ▶ IBM Director 4.10.2
- ▶ IBM Director 4.11
- ▶ IBM Director 4.12
- ▶ IBM Director 4.20
- ▶ IBM Director 4.20.2
- ▶ IBM Director 4.21
- ▶ IBM Director 4.22

Versions of IBM Director earlier than IBM Director 4.10 are incompatible with IBM Director 5.10. If you are running a version of IBM Director earlier than IBM Director 4.10, you can perform one of the following operations:

- ▶ Uninstall your existing IBM Director installation and then install IBM Director 5.10.
- ▶ Upgrade your existing IBM Director installation to a version from which you can upgrade to IBM Director 5.10. Then, upgrade to IBM Director 5.10.

5.3 Fastpath install

IBM Virtualization Engine can be compared with iSeries Navigator concerning the components. IBM Virtualization Engine 2 (VE2) is the Base Support like iSeries Navigator Base Support. IBM Director Server is an component like Basic Operations within iSeries Navigator. So you cannot run IBM Director Server without the Base Support IBM Virtualization Engine.

Before installing IBM Director, you need to review the firewalls and blocked ports in your installation environment.

Consider each of the following issues carefully when planning to install IBM Director:

- ▶ Ports 5988, 5989, and 6988 must be open in order to install IBM Director. Some firewalls might attempt to block these ports. Allow these ports to be used by the IBM Director software components.
- ▶ The Service Location Protocol (SLP) port (port 427) needs to be open if the installation environment is behind a firewall.

IBM Director Server for i5/OS is installed as part of IBM Virtualization Engine 2 Systems Edition for iSeries.

This fast path is based on the IBM Virtualization Engine 2 System Edition. It uses the installation wizard to install IBM Director Server and allow it to be integrated in a Virtualization Engine environment. The version of IBM Virtualization engine must be VE2 (5733-VE2).

Before you begin installing IBM Director Server, make sure that you meet the iSeries Hardware requirements to install IBM Director Server. You also must meet the prerequisites included in iSeries Software requirements.

To install IBM Director Server, complete the following steps:

1. Insert the **IBM Virtualization Engine Start Here installation CD** or DVD into the CD or DVD device on a Windows system that has a network connection to the iSeries system on which you plan to install IBM Director Server. The installation wizard starts automatically if auto run is set within your windows environment. If this is not the case, just open explorer within Windows and click your CD-Drive, then double-click **autorun.bat**. The **..installVEi5OS.exe** dos prompt opens.

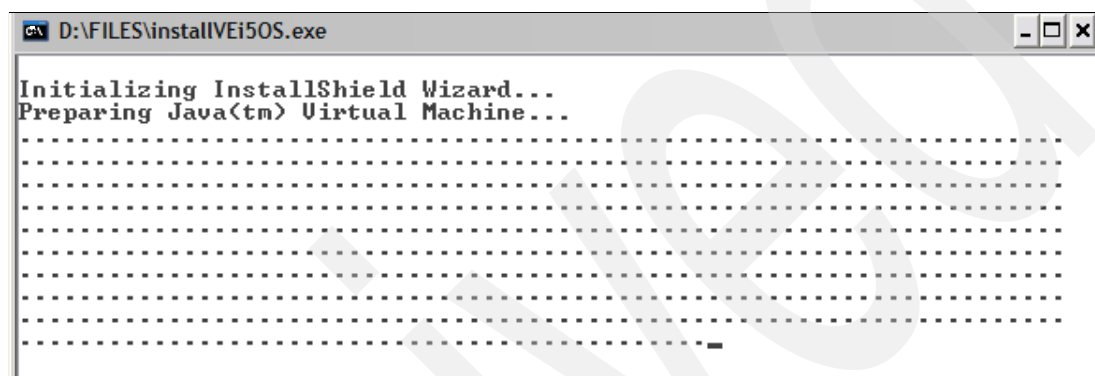


Figure 5-4 *installVEi5OS.exe dos prompt*

2. On the i5/OS Sign-On page (Figure 5-5), specify the following information:
 - a. In the **Server Name** field, type the fully qualified server name or IP Address on which you want to install IBM Director Server.
 - b. In the **User ID** field, type the user ID to sign on to the system on which you are installing IBM Director Server, we advise you use QSECOFR.
 - c. In the **Password** field, type the password associated with the user ID.



Figure 5-5 *i5OS Sign-On*

3. On the Welcome window (Figure 5-6 on page 116), click **Next**.

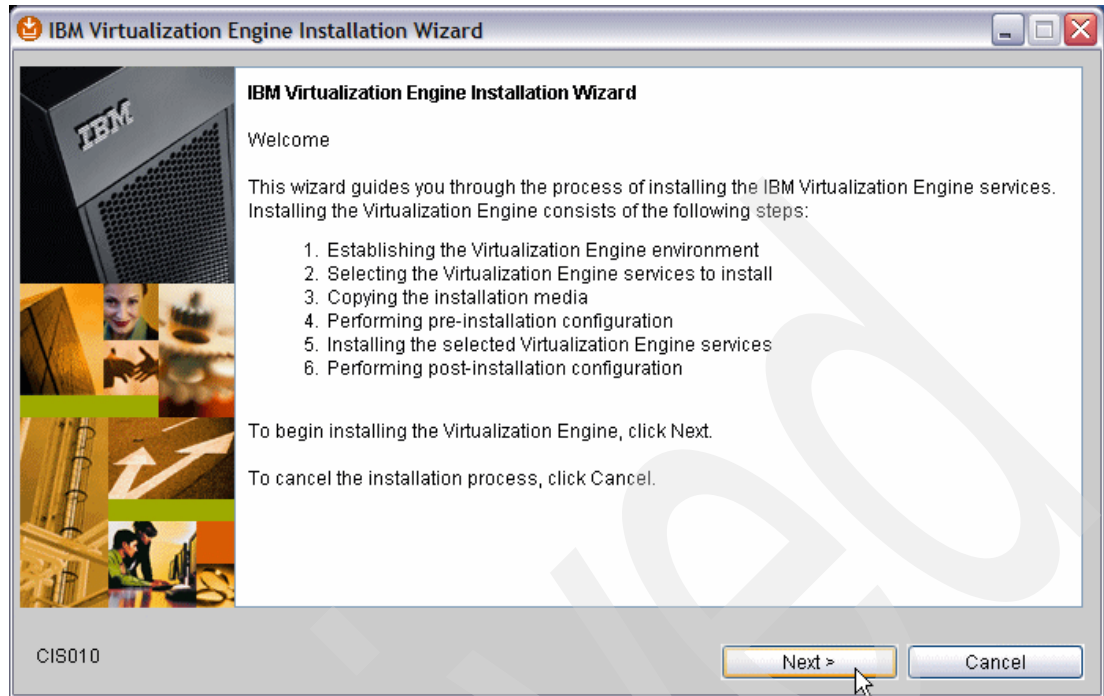


Figure 5-6 Welcome window

4. On the Software Agreements window (Figure 5-7), select the **IBM Director with Virtualization Engine Console** button and read the software license agreements. If you agree with the terms, select **I accept the terms of the license agreements** and click **Next**. Otherwise, you will need to cancel the installation.



Figure 5-7 Software Agreements window, accept the terms of the license agreements

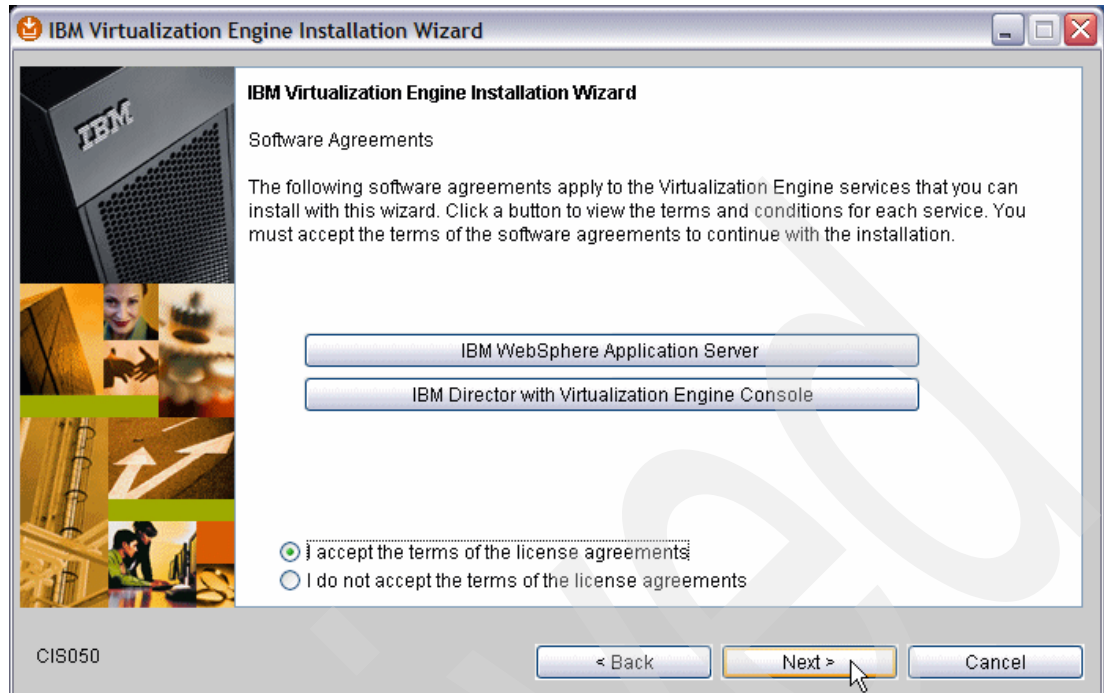


Figure 5-8 Software Agreements window, Next

5. On the Temporary Directory window (Figure 5-9), the default directory is displayed to indicate where the installation files are copied. The default directory is C:\ProgramFiles\IBM\VE2i5OS. To select a different location for storing installation files, click **Browse**. Or just click **Next**.

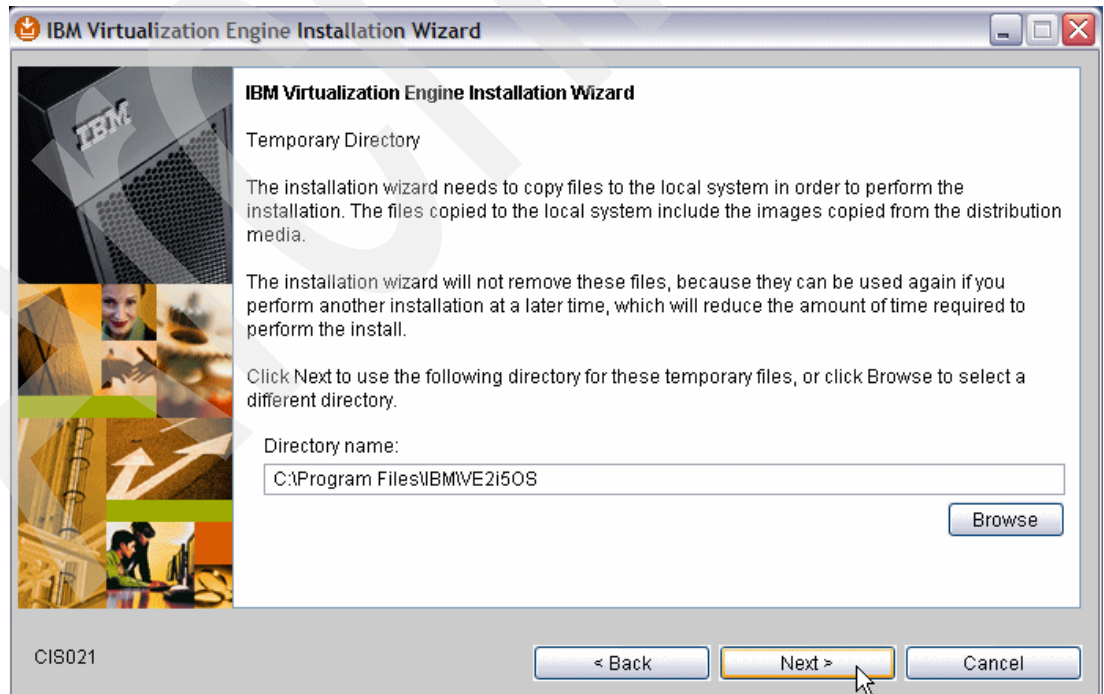


Figure 5-9 Temporary Directory window

6. On the Log File Destination window (Figure 5-10), the default path for the log-file destination is displayed. The default path is /QIBM/UserData/VE2/Logs. You cannot edit the destination directory for the log files. Notice this path because in this path the install logs appear, and if the install fails for some reason, these log files should be checked. Just click **Next**.

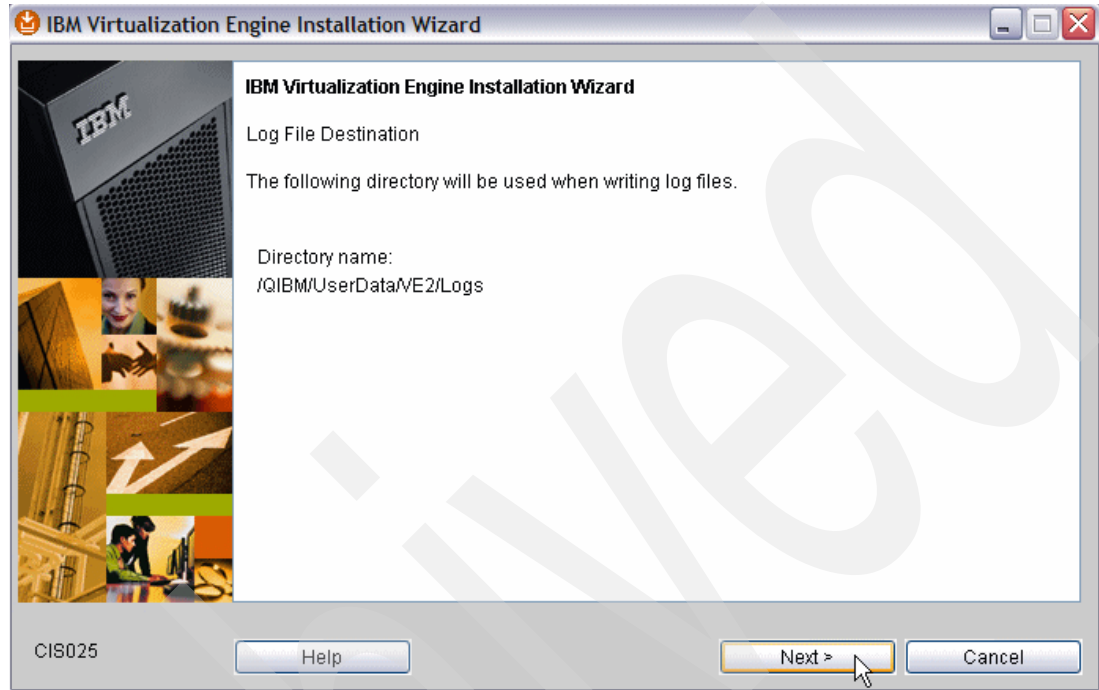


Figure 5-10 Log File Destination window

7. On the Service Selection window (Figure 5-11 on page 119), select **IBM Director Server** and click **Next**.

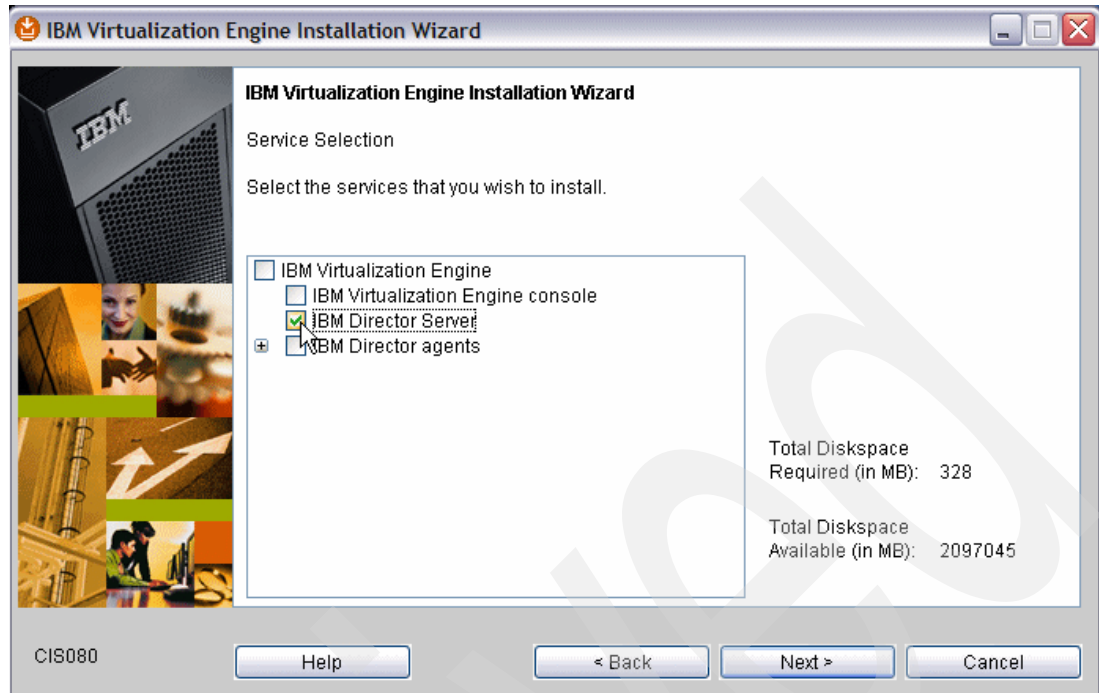


Figure 5-11 Service Selection window, select IBM Director Server

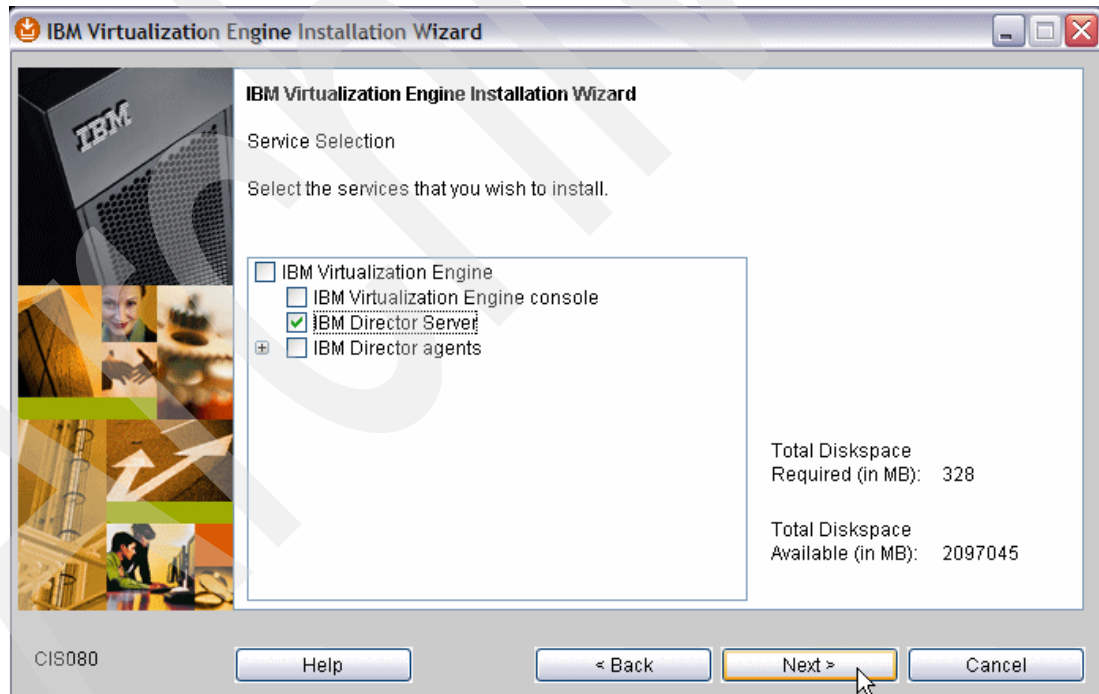


Figure 5-12 Service Selection window, click Next

8. On the Service Selection Summary window (Figure 5-12), verify your selections. If the summary is correct, click **Next**.

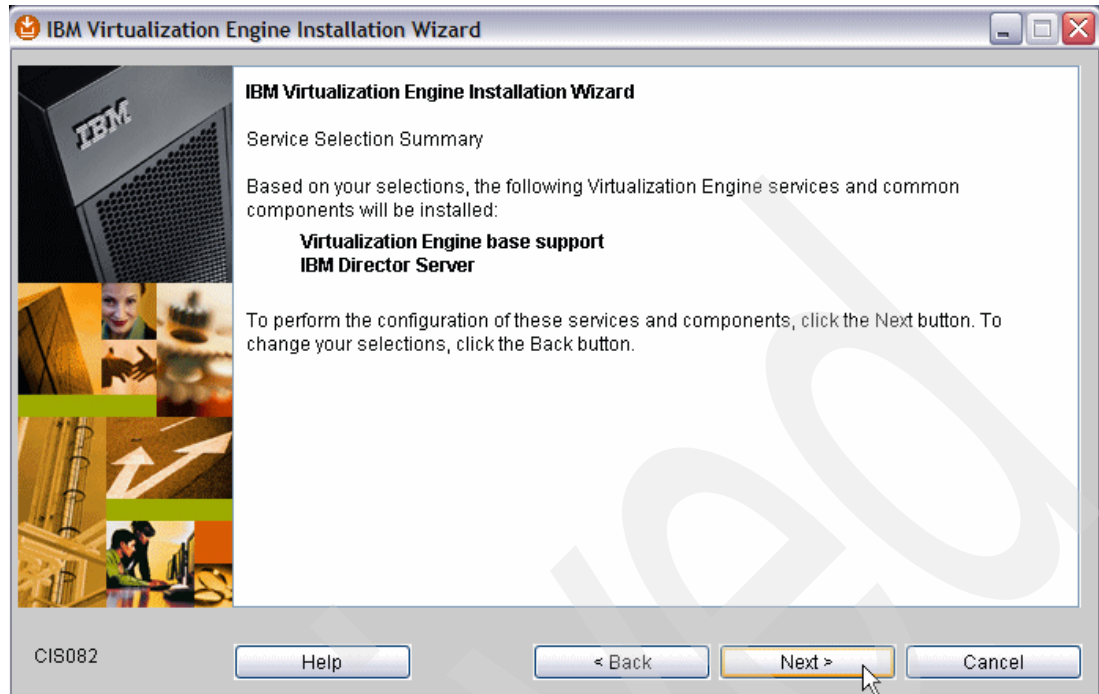


Figure 5-13 Service Selection Summary window

9. An Installation Wizard Restart window (Figure 5-14) opens, wait a few moments.

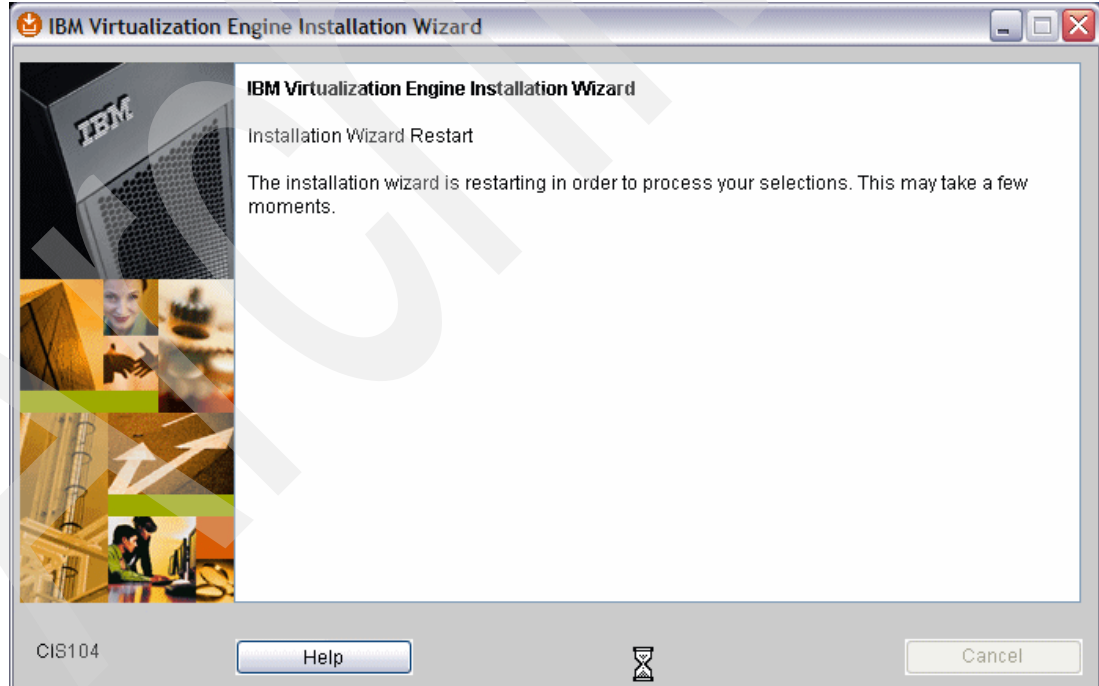


Figure 5-14 Installation Wizard Restart window

10. On the Media Copy window (Figure 5-15 on page 121), insert the second CD-ROM listed on the window and follow the instructions and click **Copy Media**. The installation files will be copied for IBM Virtualization Engine Common Components to the temporary directory.

Note: The Media Copy window only appears when the installation files are not on the PC. If the installation files are already on the PC, a window opens as in Figure 5-19 on page 123.

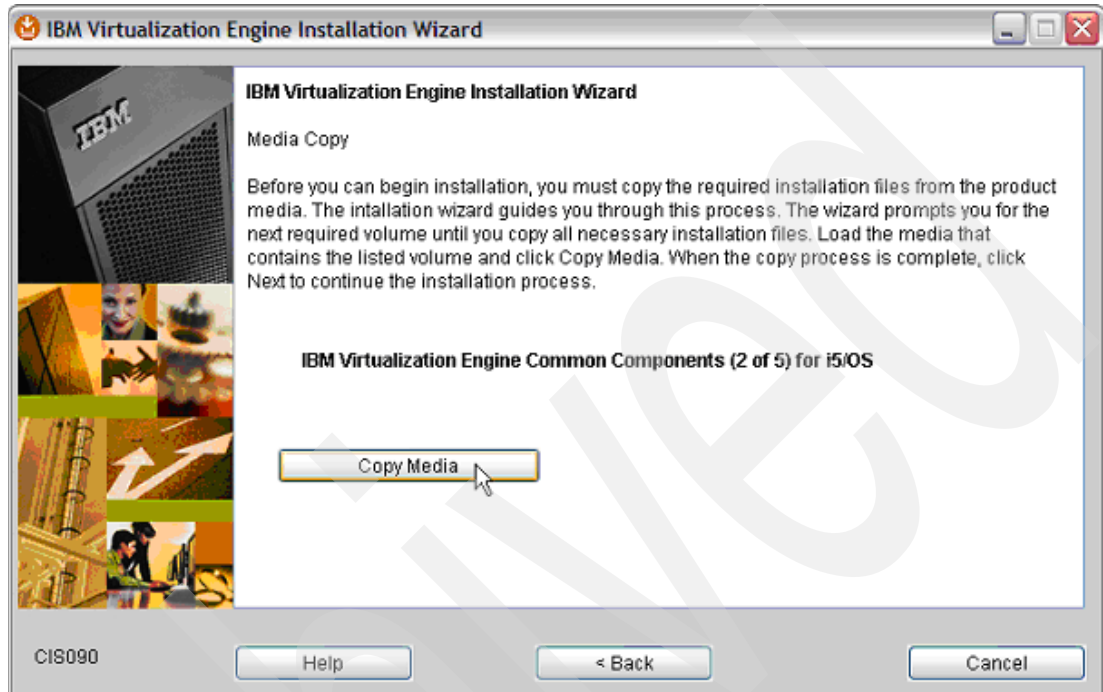


Figure 5-15 Media Copy window

11.A Please Wait window (Figure 5-16 on page 122) opens to indicate that the installation files are being copied from the installation media to the temporary directory.

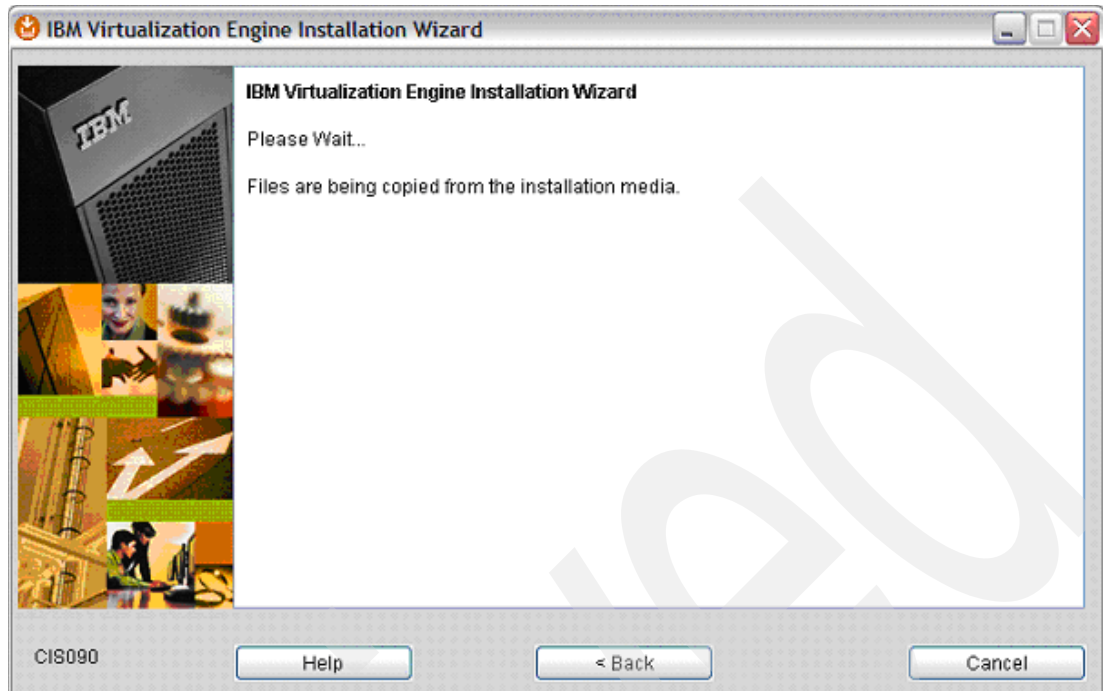


Figure 5-16 Please Wait window

12. Another Media Copy window (Figure 5-17) opens to load the Next Volume. Follow the instructions and insert the specified CD-ROM on your Windows workstation to copy the required IBM Director 5.10 installation files to the temporary directory.

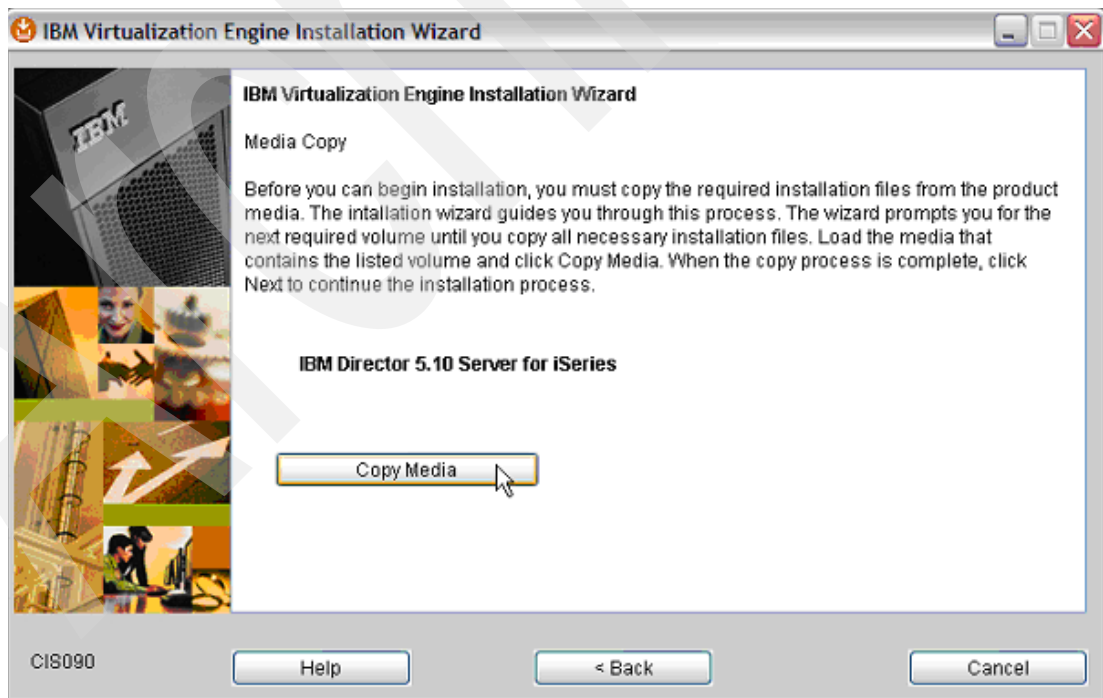


Figure 5-17 Media Copy window

13. A Please Wait window (Figure 5-18 on page 123) opens to indicate that the installation files are being copied from the installation media to the temporary directory.

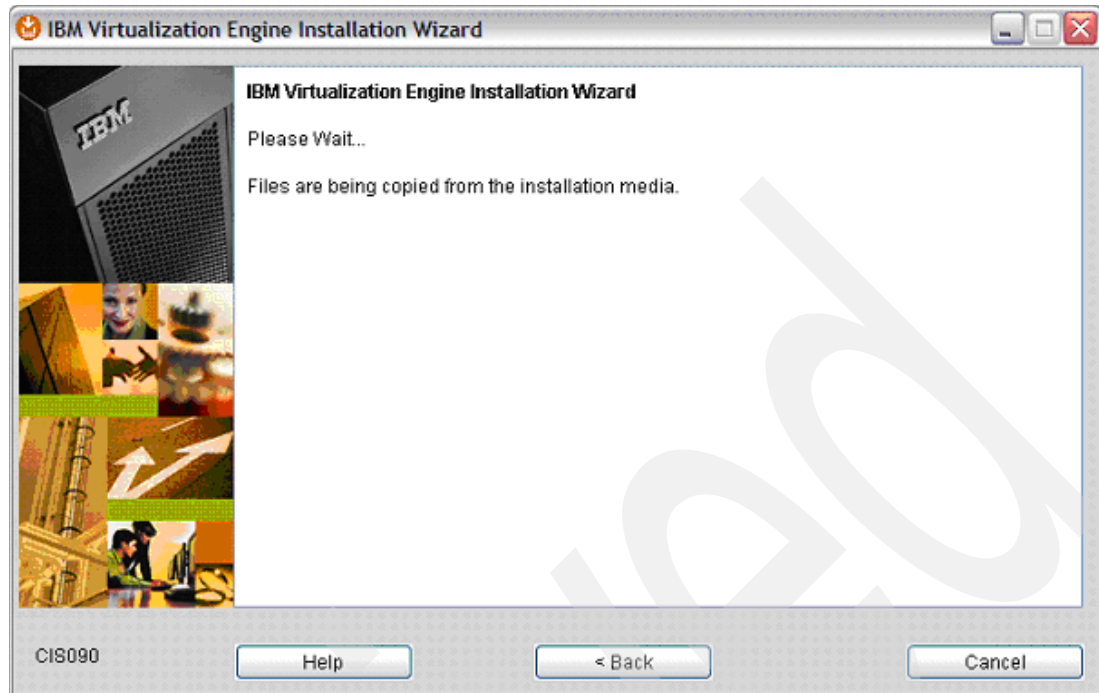


Figure 5-18 Please Wait window

14. Another Media Copy window (Figure 5-19) opens indicating that no more media is needed for the install. Click **Next**. You will get this window at step 10, if you have already got the installation files on your Windows workstation. The files are in the temporary directory specified in step 5.

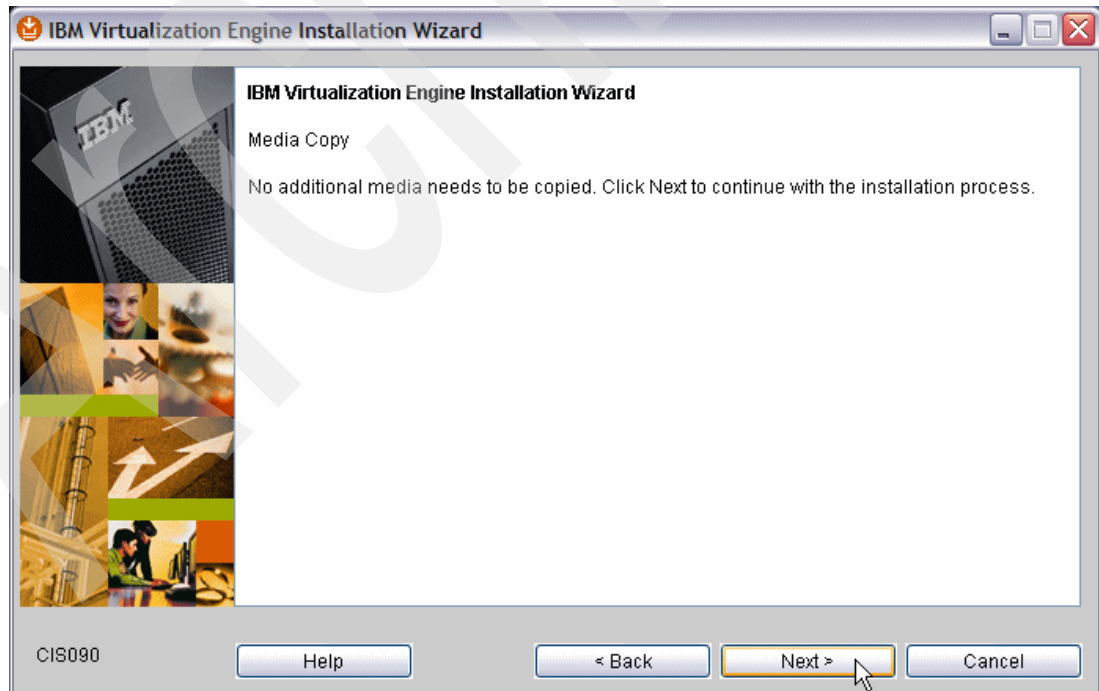


Figure 5-19 Media Copy window

15. An Installation Wizard Restart window (Figure 5-20) opens again, wait a few moments.

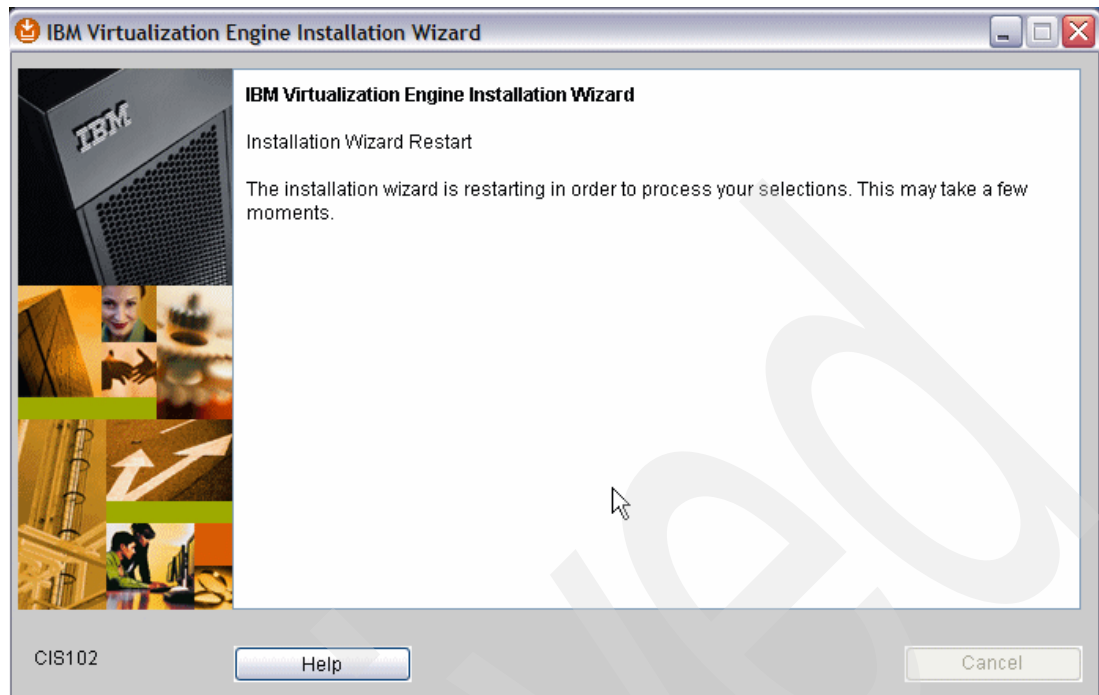


Figure 5-20 Installation Wizard Restart window

16. The Selection Summary window (Figure 5-21) opens to indicate the services that are selected for installation. If you need to make changes, click **Back**. To continue with the installation, click **Install**.

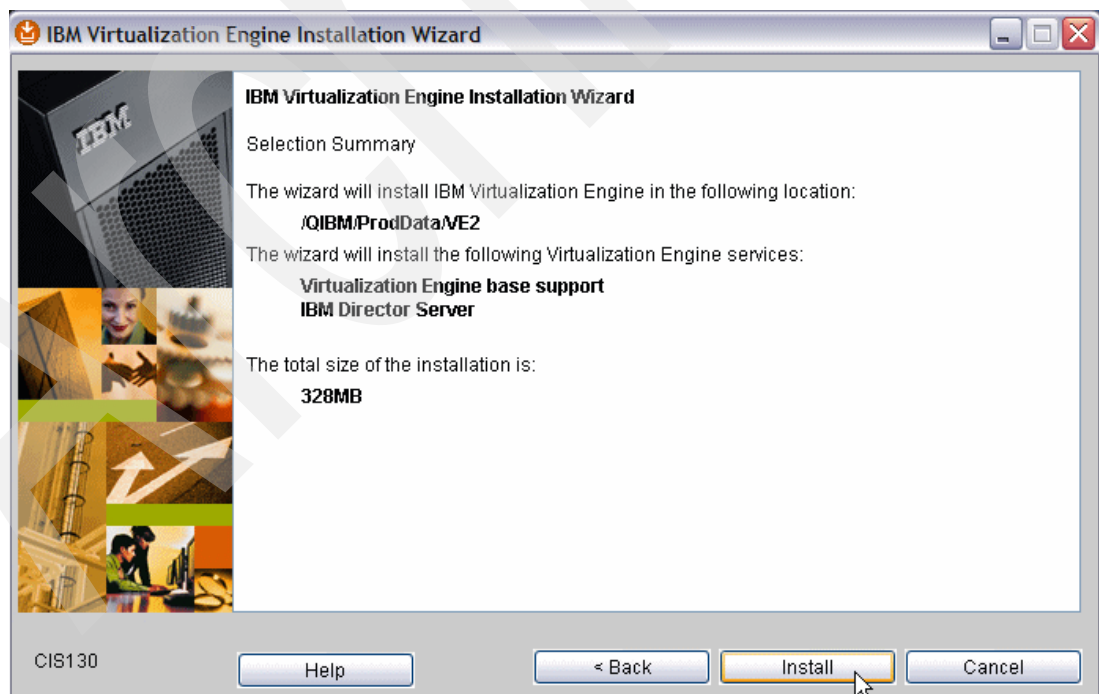


Figure 5-21 Selection Summary window

17. An Installation Progress window (Figure 5-22) opens to indicate the installation progress for IBM Virtualization Engine base Support.

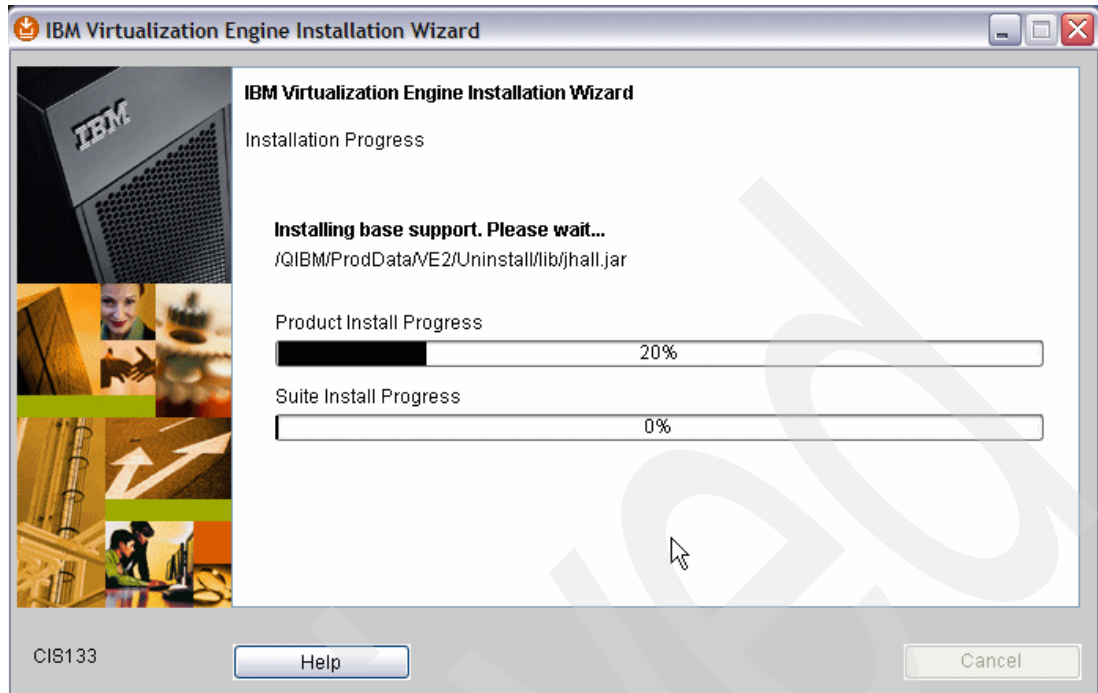


Figure 5-22 Installation progress window IBM Virtualization Engine Base Support

18. An Installation Progress window (Figure 5-23) opens to indicate the installation progress of creating the uninstaller for IBM Virtualization Engine.

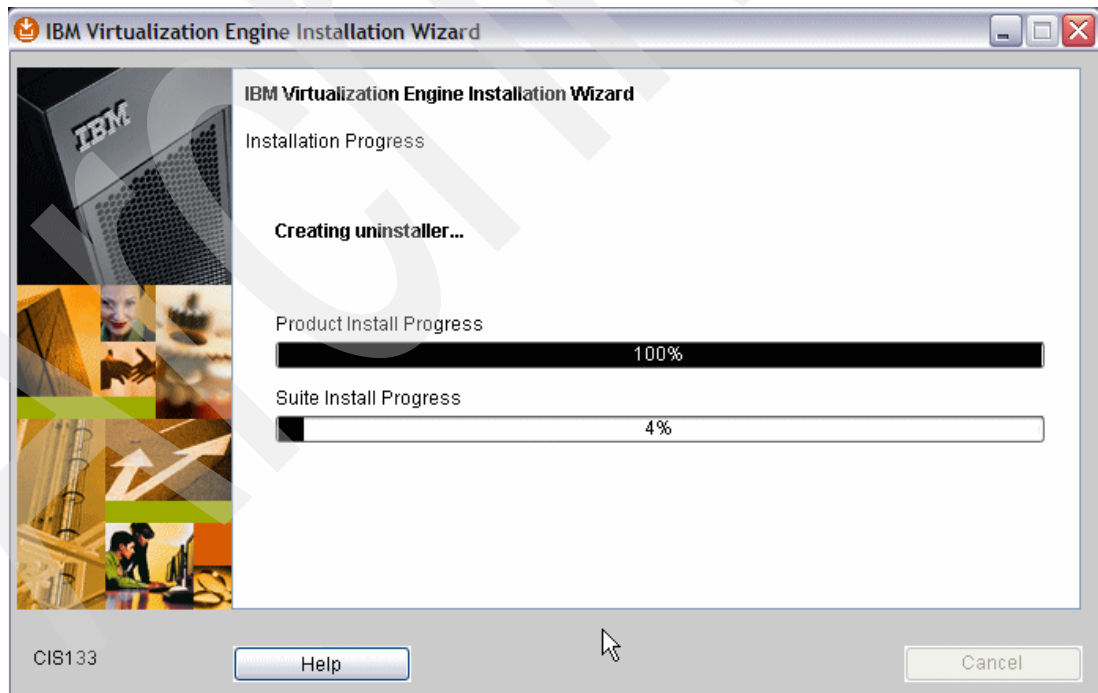


Figure 5-23 Installation progress window uninstaller IBM Virtualization Engine Base Support

19. An Installation Progress window (Figure 5-24) opens to indicate the installation progress for IBM Director Server.

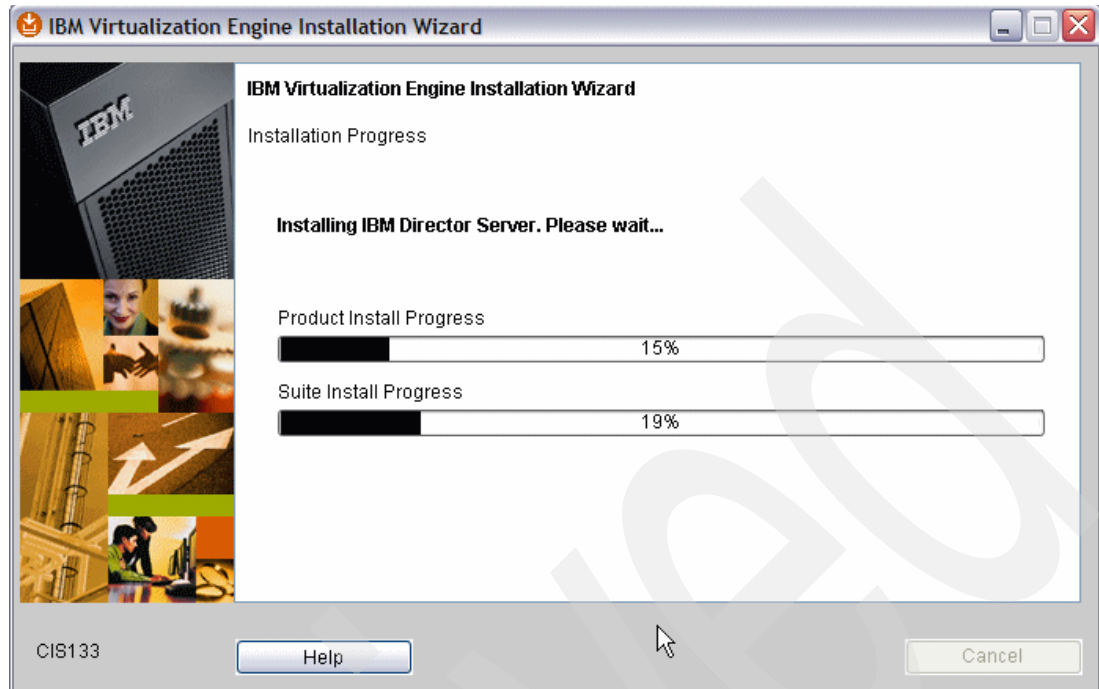


Figure 5-24 Installation progress window IBM Director Server

20. An Installation Progress window (Figure 5-25) opens to indicate the installation progress of creating the uninstaller for IBM Director Server.

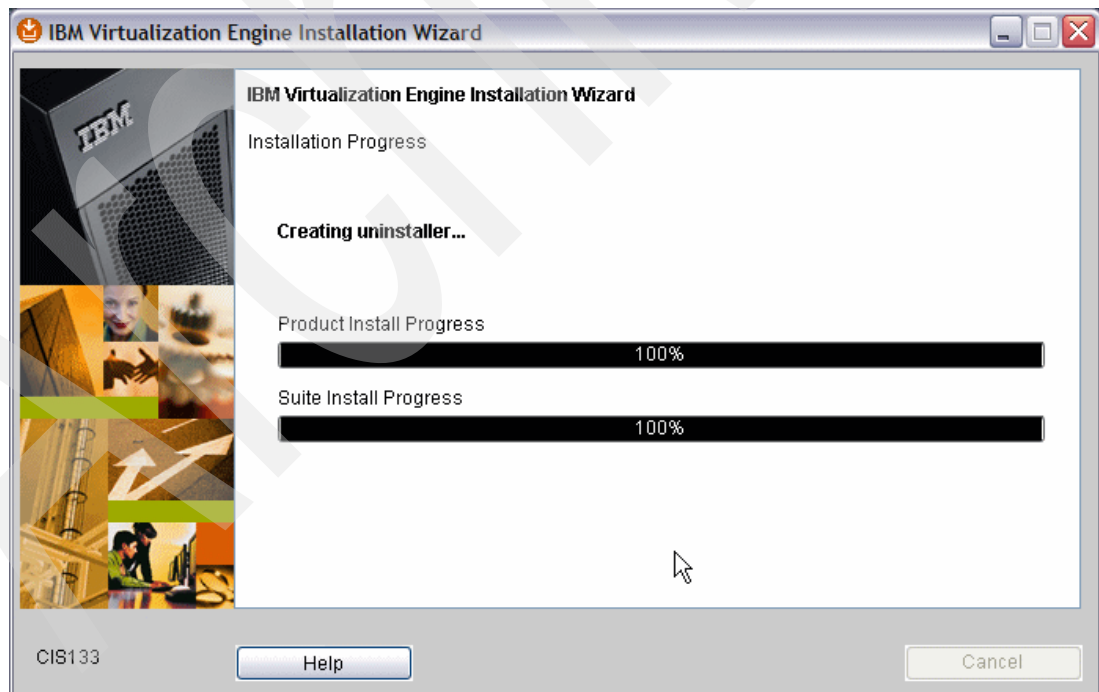


Figure 5-25 Installation progress window uninstaller IBM Director Server

21. On the Server Start Preference window (Figure 5-26), select **Start IBM Director** to indicate to start IBM Director Server when the installation is complete. Click **Next**. This

does not mean the IBM Director Server starts automatically with TCP/IP. More information about this is in 5.2.2, “Start IBM Director Server” on page 109.

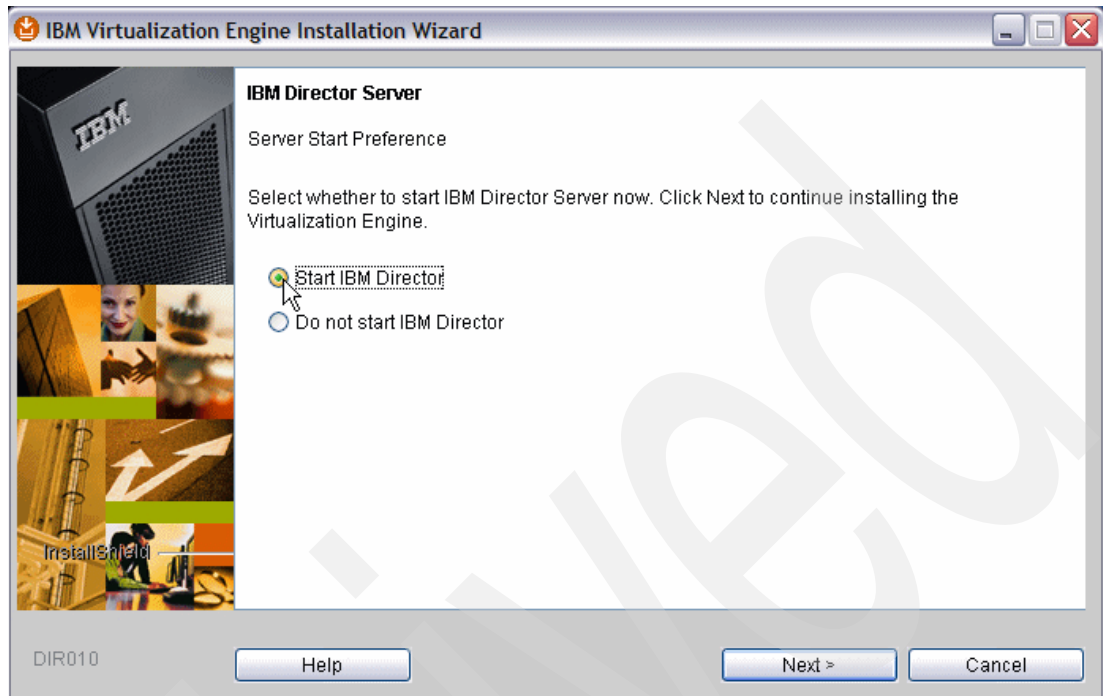


Figure 5-26 Server Start Preference window: Select Start IBM Director

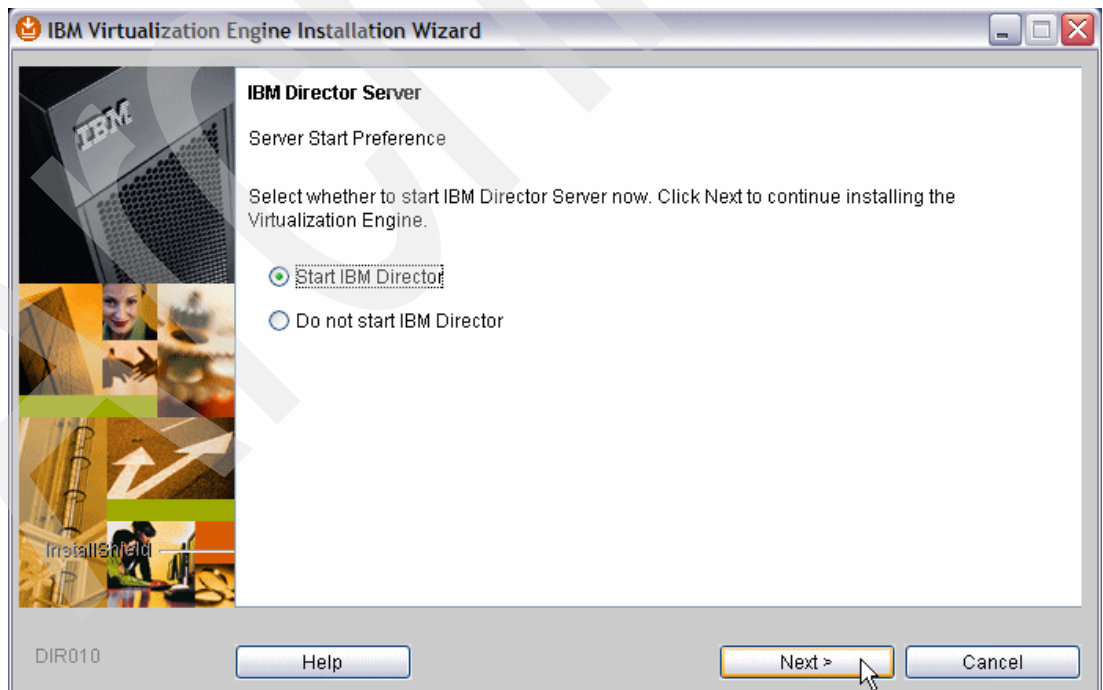


Figure 5-27 Server Start Preference window: Select Next

A Service Registration window (Figure 5-28) opens, just click **Next**. The sentence “One or more components failed to register successfully” can be ignored if you are just installing IBM Director Server.

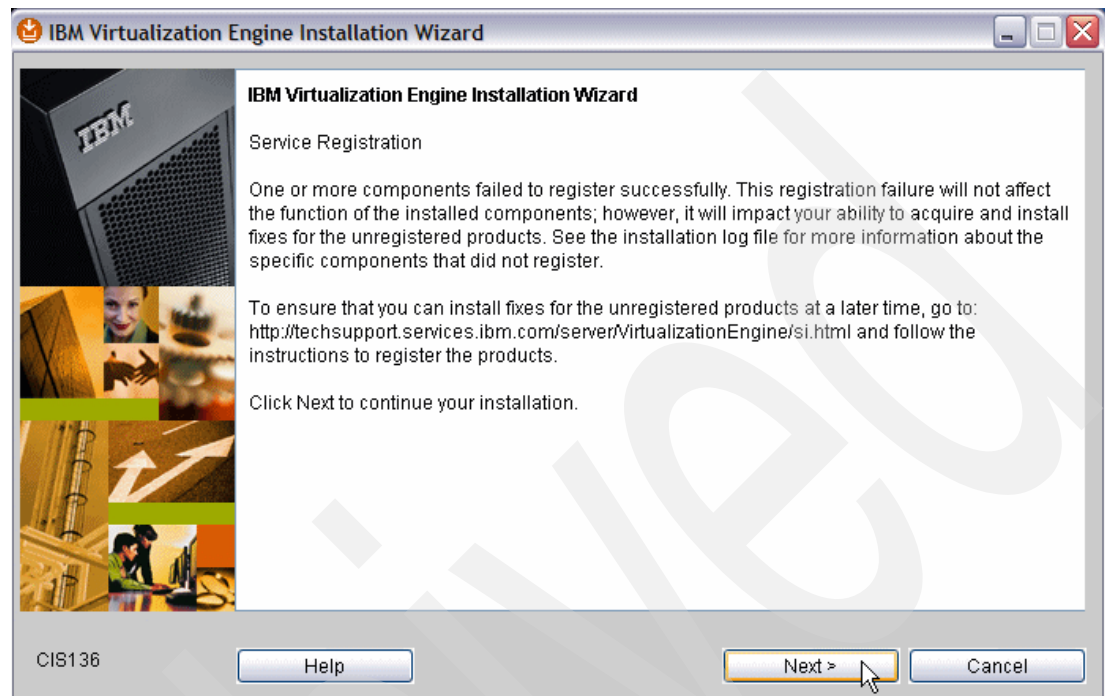


Figure 5-28 Service Registration window

22.A Installation Summary window (Figure 5-29) opens that summarizes the results of the installation. Click **Finish**.

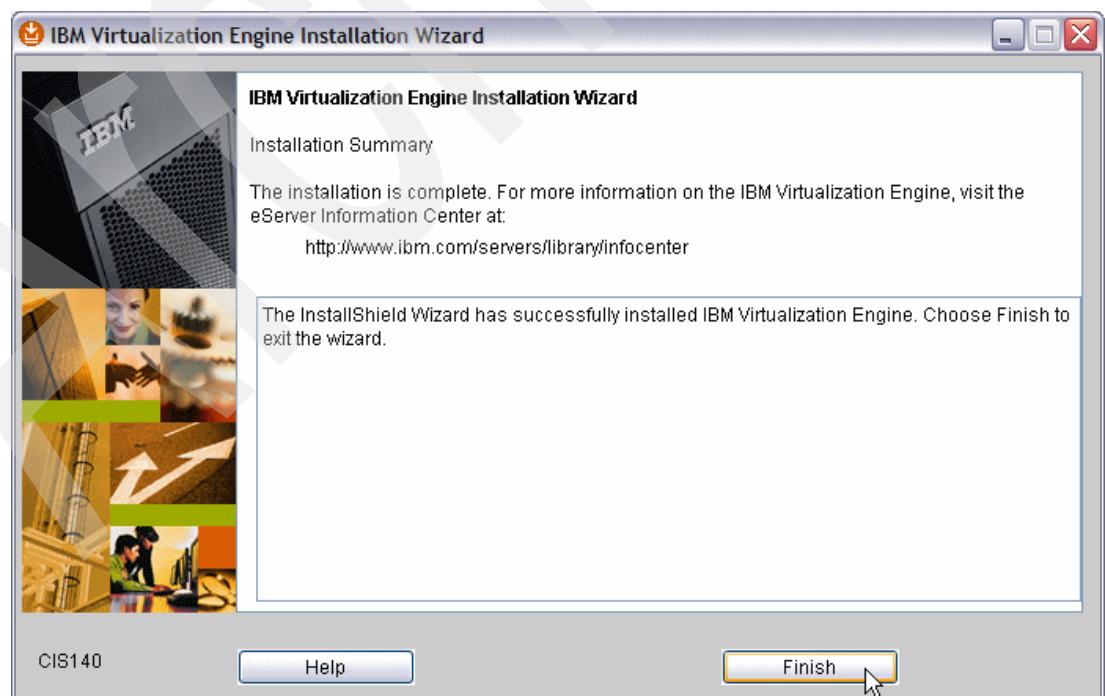


Figure 5-29 Installation Summary window

5.3.1 Uninstall IBM Director Server on i5/OS using the Uninstaller Launchpad

Note: To uninstall IBM Director Server, you must connect to the server running i5/OS from a system running Windows. The system running Windows must have JRE™, Version 1.4 or later installed.

If you want to uninstall IBM Director Server for some reason, you should do this by using the Uninstaller Launchpad method. Complete the following steps to uninstall IBM Director Server running on i5/OS:

1. On the Windows workstation that you used for installing Virtualization Engine software on your i5/OS system, map a network drive to the following folder: `\\system-name\QVE2` where system-name is the name of the i5/OS system from which you want to uninstall one or more Virtualization Engine products. This mapped drive creates a mapping to the /QIBM/ProdData/VE2 directory that contains the Uninstaller Launchpad executable file. The procedure to map the drive is as follows:
 - a. Open **Windows Explorer** on your Windows Workstation with a connection to the i5/OS system.
 - b. Select **Tools** → **Map Network Drive**.

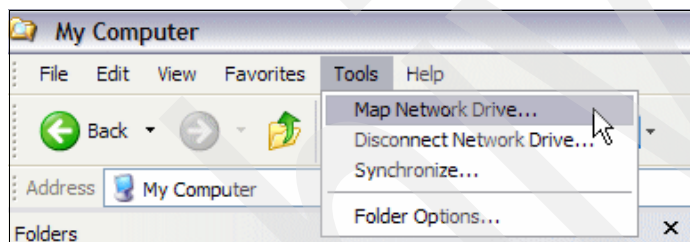


Figure 5-30 Map Network Drive... window

- c. The Map Network Drive window opens.
 - i. Select a free drive letter from the pull-down menu.
 - ii. Type the folder name as follows: `\\yoursystemname\QVE2`
 - iii. Click **Finish**.

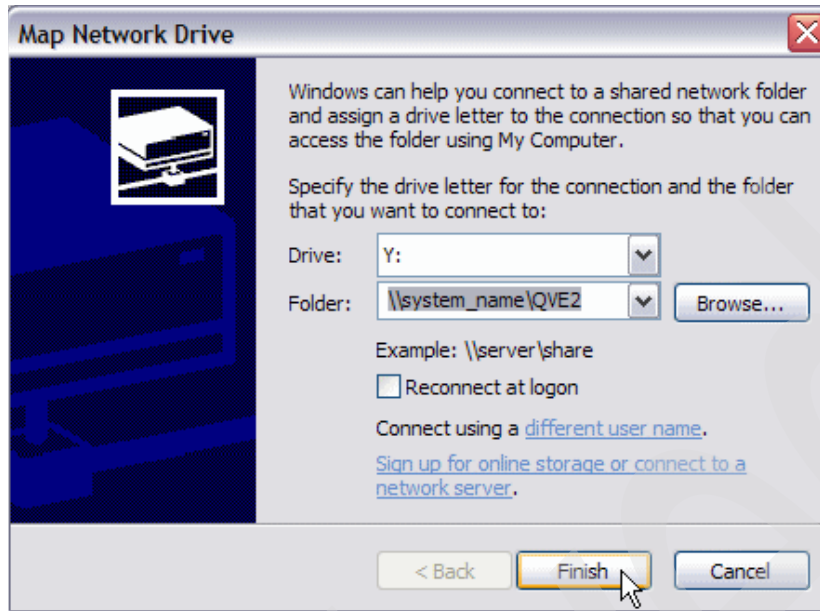


Figure 5-31 Map Network Drive window

- d. If you are logged on your Windows workstation with an user ID which does not exist on the 5/OS system, you have to connect using a different user name as follows, otherwise skip this step.
 - i. Select **Connecting using a different user name**. A Connect As window opens.
 - ii. Type an existing i5/OS User name and password like shown below.
 - iii. Click **OK**.
 - iv. It returns to the previous Map network Drive window. Click **Finish**.

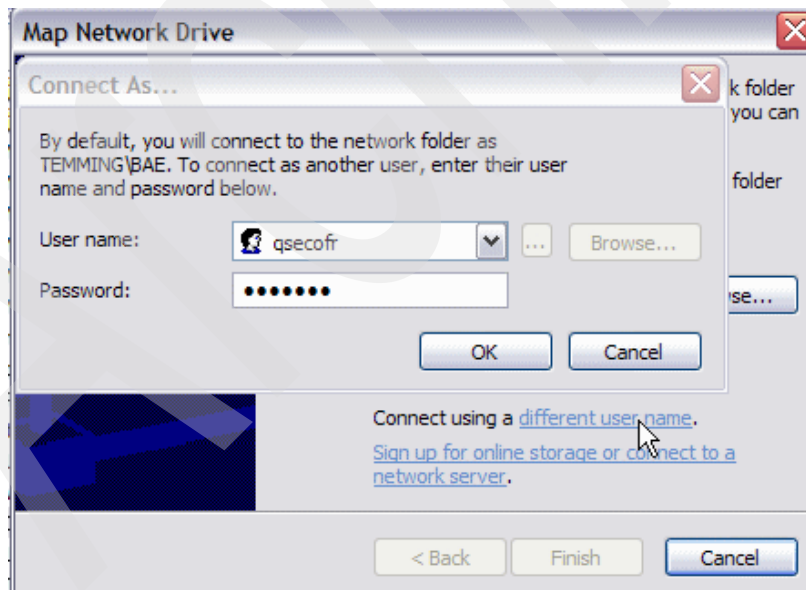


Figure 5-32 Connect using a different user name window

2. Open a Windows command prompt, change to this mapped drive letter. For example, if the mapped drive is Y, type Y: and press Enter. Then type Dir and press Enter. Verify that there is a file named uninstallVEi5OS.bat in the directory.

```

C:\Documents and Settings\Administrator>y:
Y:\>dir
Volume in drive Y has no label.
Volume Serial Number is 0000-0000

Directory of Y:\

01/19/2006  10:24    <DIR>        .
01/19/2006  10:12    <DIR>        ..
01/19/2006  10:23    <DIR>        META-INF
01/19/2006  10:24             1,354 UEInstall.properties
01/19/2006  09:36    <DIR>        license
01/19/2006  09:37    <DIR>        ManagedNodes
01/19/2006  09:38             30,226,013 si_setup.jar
01/19/2006  10:23             1,154 vefixlul.xml
01/19/2006  09:49    <DIR>        Uninstall
07/29/2005  12:58             1,402 uninstallVEi5OS.bat
01/19/2006  09:54    <DIR>        lib
01/19/2006  09:49    <DIR>        properties
01/19/2006  09:49    <DIR>        rpd
01/19/2006  09:54    <DIR>        bin
01/19/2006  09:54    <DIR>        gmi
01/19/2006  09:55    <DIR>        _uninst
01/19/2006  10:22    <DIR>        Director
01/19/2006  10:24             447 UEULResponsefile.dat
                    5 File(s)      30,230,370 bytes
                   13 Dir(s)  267,563,470,848 bytes free

Y:\>_

```

Figure 5-33 DOS Prompt showing uninstallVEi5OS.bat

- To start the Uninstaller Launchpad, type the following command and press Enter:
uninstallVEi5OS.bat mapped_drive: [system-name:userID:password] where
mapped_drive is the letter of the mapped drive to the QVE2 share, and
[system-name:userID:password] is the i5/OS host name, i5/OS user profile, and password
for the user profile separated by the colon (:). So, for example:

uninstallVEi5OS.bat Y: demo.ibm.com:qsecofr:secofr1

```

C:\Documents and Settings\Administrator>y:
Y:\>dir
Volume in drive Y has no label.
Volume Serial Number is 0000-0000

Directory of Y:\

01/19/2006  10:24    <DIR>        .
01/19/2006  10:12    <DIR>        ..
01/19/2006  10:23    <DIR>        META-INF
01/19/2006  10:24             1,354 UEInstall.properties
01/19/2006  09:36    <DIR>        license
01/19/2006  09:37    <DIR>        ManagedNodes
01/19/2006  09:38             30,226,013 si_setup.jar
01/19/2006  10:23             1,154 vefixlul.xml
01/19/2006  09:49    <DIR>        Uninstall
07/29/2005  12:58             1,402 uninstallVEi5OS.bat
01/19/2006  09:54    <DIR>        lib
01/19/2006  09:49    <DIR>        properties
01/19/2006  09:49    <DIR>        rpd
01/19/2006  09:54    <DIR>        bin
01/19/2006  09:54    <DIR>        gmi
01/19/2006  09:55    <DIR>        _uninst
01/19/2006  10:22    <DIR>        Director
01/19/2006  10:24             447 UEULResponsefile.dat
                    5 File(s)      30,230,370 bytes
                   13 Dir(s)  267,563,470,848 bytes free

Y:\>uninstallVEi5OS.bat Y: demo.ibm.com:qsecofr:secofr1_

```

Figure 5-34 DOS prompt execute the uninstall command

Note: Because you are uninstalling IBM Director from a remote i5/OS system, the Uninstaller Launchpad might take a few minutes to display and run, depending on network connection speed.

4. On the IBM Virtualization Engine Uninstaller Launchpad (Figure 5-35), select **IBM Director Server** in the list of products to uninstall and click **Uninstall**.

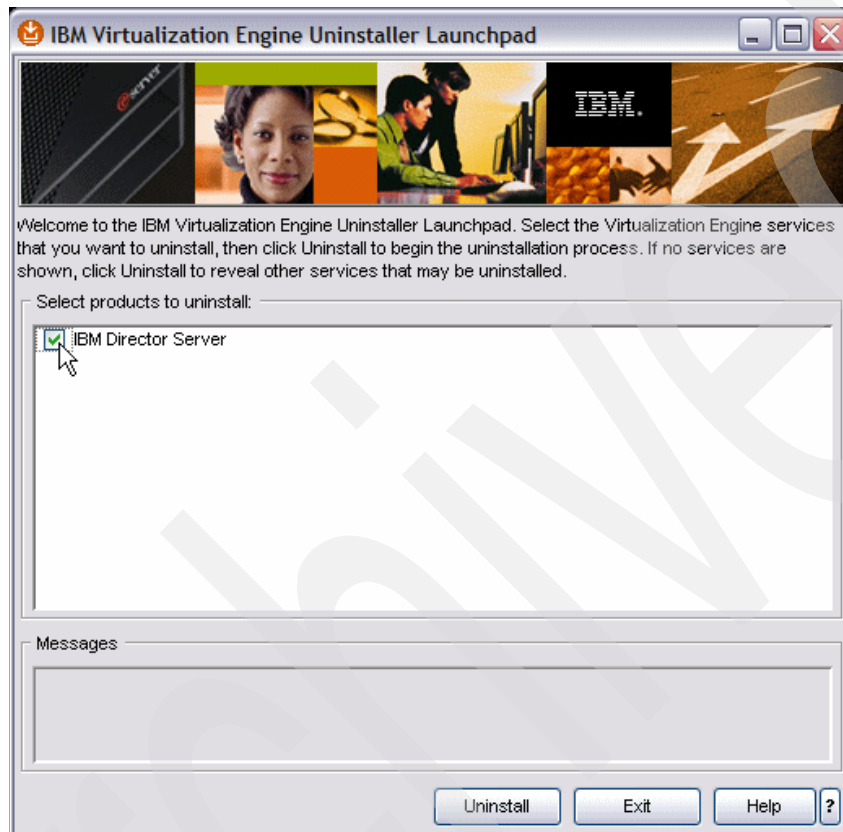


Figure 5-35 IBM Virtualization Engine Uninstaller Launchpad select IBM Director Server

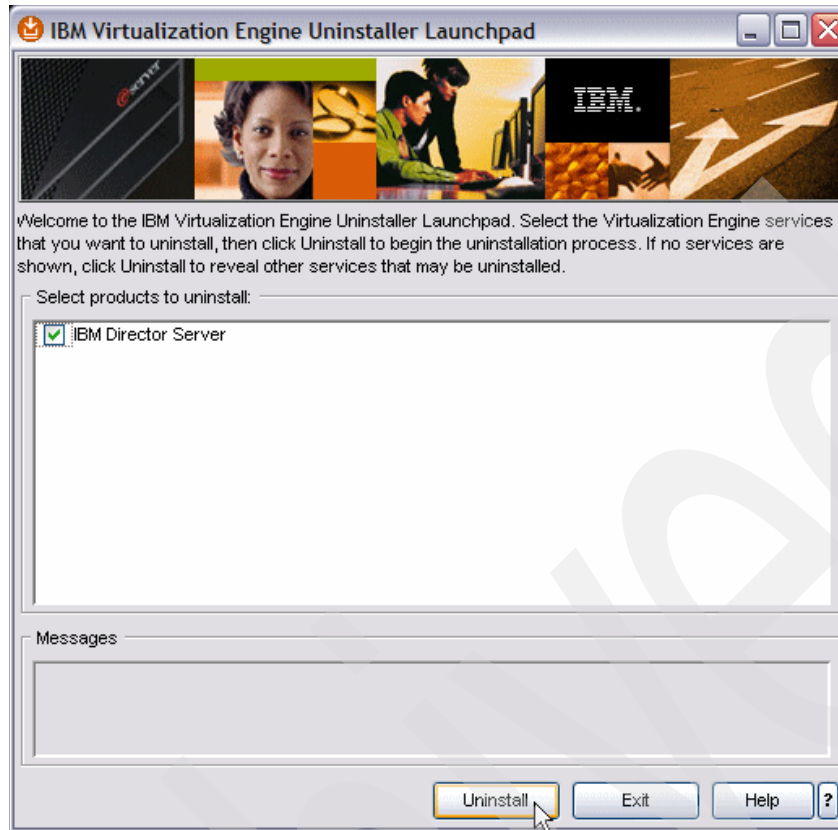


Figure 5-36 IBM Virtualization Engine Uninstaller Launchpad: Select Uninstall

5. The IBM Virtualization Engine Uninstaller Launchpad window refreshes and indicates the uninstall has started.

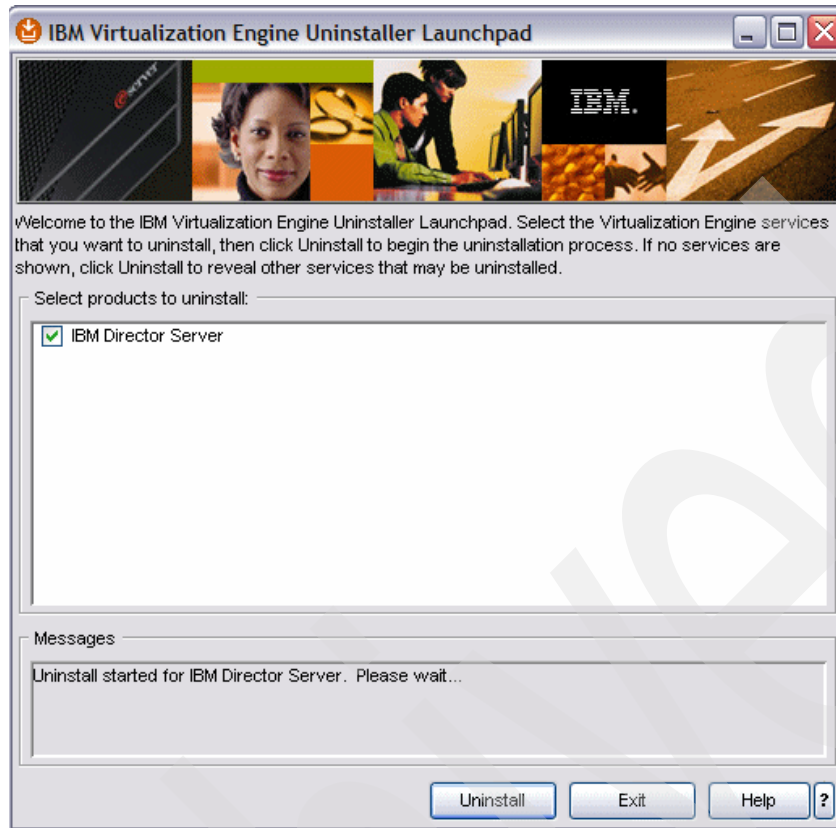


Figure 5-37 IBM Virtualization Engine Uninstaller Launchpad Uninstall started

6. When the uninstall process for IBM Director Server is complete, it displays it in the *Messages area*. And a new window opens asking the question if you want to uninstall IBM Virtualization Engine Base Components. Select **Yes** if you want to do this; otherwise, select No.

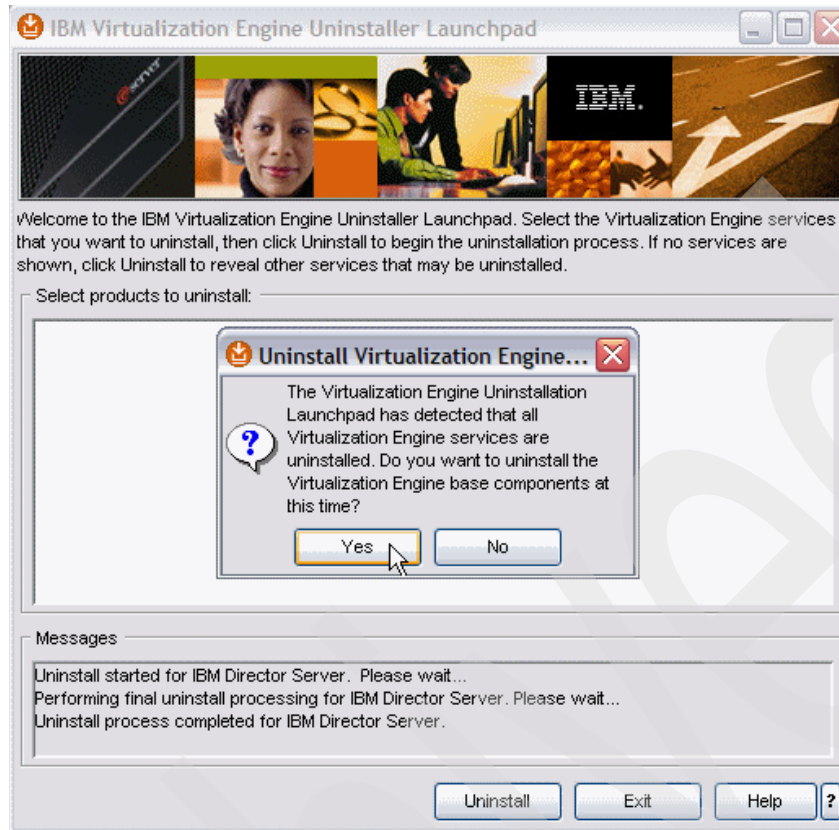


Figure 5-38 Uninstall IBM Virtualization Engine Base Components

7. To remove IBM Director user data from the i5/OS management server, delete the /qibm/userdata/director/ directory using the following steps in QSHHELL:
 - a. On an 5250 emulation command line, type **strqsh** and press Enter.
 - b. The QSH Command entry display opens, type the following on the command line and press Enter:


```
rm -rf /qibm/userdata/director/
```
 - c. On the display, the results of the command appears. When the dollar sign character(\$) returns, the command is finished.

5.3.2 Uninstall IBM Director Server using DLTICPGM on i5/OS

If you do not operate in an IBM Virtualization Engine environment, you can uninstall IBM Director Server using DLTICPGM. If you do run in IBM Virtualization Engine environment, you should not follow these steps but follow the ones described in 5.3.1, “Uninstall IBM Director Server on i5/OS using the Uninstaller Launchpad” on page 129.

To uninstall IBM Director Server using DLTICPGM, complete the following steps:

1. At an i5/OS command prompt on the system on which IBM Director Server is installed, type the following command and press Enter:


```
DLTICPGM LICPGM(5722DR1)
```
2. To remove IBM Director user data from the i5/OS management server, delete the /qibm/userdata/director/ directory using the following steps in QSHHELL:
 - a. On an 5250 emulation command line, type **strqsh** and press Enter.

- b. The QSH Command entry display opens, type the following on the command line and press Enter:

```
rm -rf /qibm/userdata/director/
```
- c. On the display, the results of the command appears. When the dollar sign character (\$) returns, the command is finished.

Important: If you installed IBM Director Server using the Virtualization Engine installation wizard, but choose to uninstall IBM Director Server using DLTLCIPGM, the configuration change will not be updated in the Global Configuration Repository. This is only the case when you have IBM Director Console installed as well. If you have only IBM Director Server installed there is no problem to use the above procedure, because then there is no Global Configuration Repository.

5.4 For Install failures

If you encounter installation problems of IBM Virtualization Engine Base Support or IBM Director Server, you can check two installation log files. These are placed in the integrated file system as indicated at step 6 in 5.3, “Fastpath install” on page 114.

The path is as follows where *<VE Log Folder>* is equal to */qibm/userdata/VE2/logs*:

- ▶ *<VE Log Folder>/installLog.txt*// contains error messages
- ▶ *<VE Log Folder>/installDebug.txt*// contains debug trace

5.4.1 Steps to display install log files using iSeries Navigator

To display log files (Figure 5-39 on page 137):

1. Open **iSeries Navigator**.
2. Expand **yourservername**.
3. Expand **File Systems** → **Integrated File Systems** → **Root** → **QIBM** → **UserData** → **VE2**.
4. Click **Logs** at the left pane.

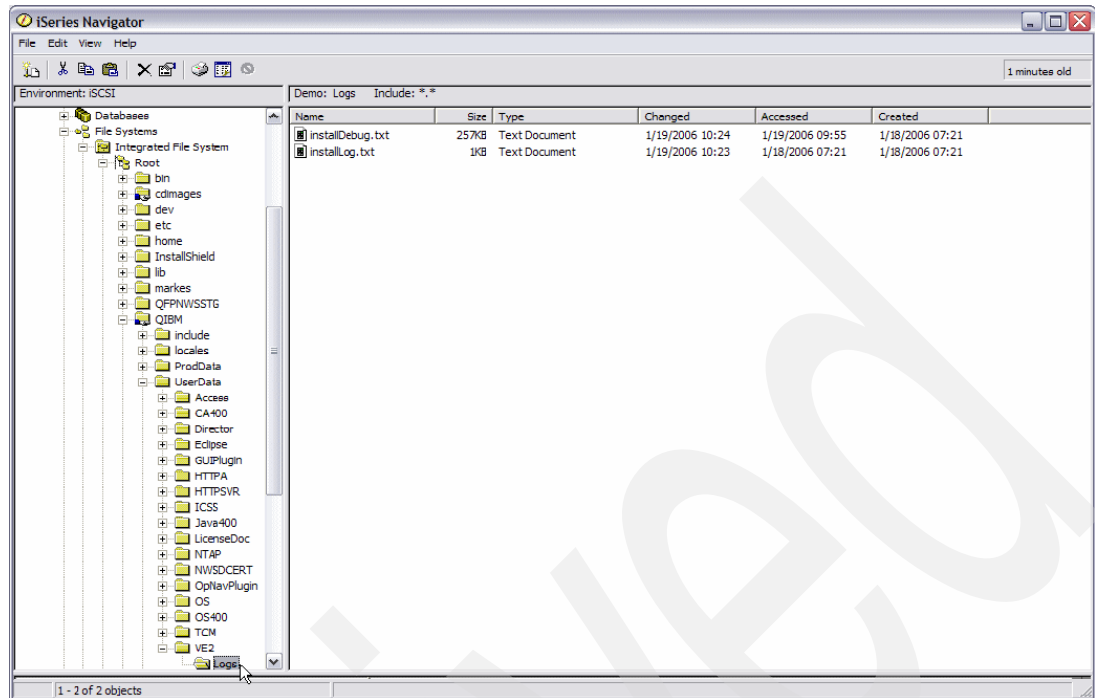


Figure 5-39 Install Log files IBM Virtualization Engine

5.4.2 Display install log files using CL command line

Follow these steps:

1. Open an 5250 emulation window.
2. Type the following command on the command line:

```
wrk1nk '/QIBM/UserData/VE2/Logs/*'
```

3. The Work with Object Link window (Figure 5-40) appears showing the log files. Enter option 5 (Display) in front of them to display the contents.

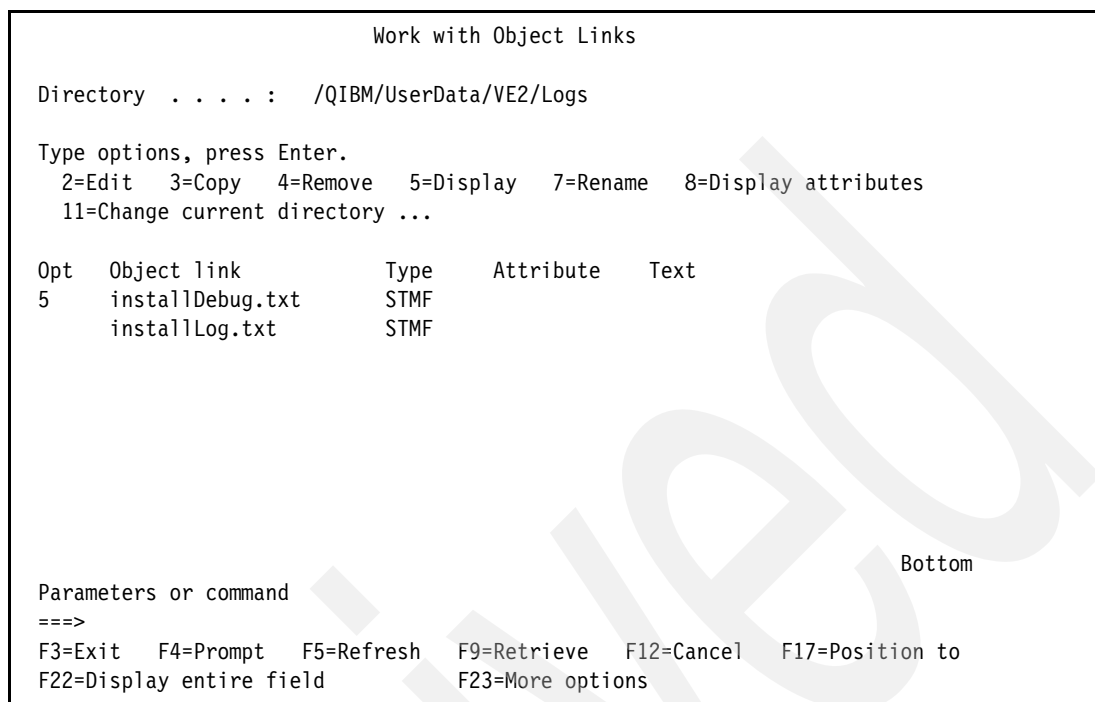


Figure 5-40 WRKLNK CL command

5.5 For uninstall failures

If you encounter problems during the Uninstall of IBM Virtualization Engine or IBM Director Server using the Uninstaller Launchpad as described in 5.3.1, "Uninstall IBM Director Server on i5/OS using the Uninstaller Launchpad" on page 129, there are some log files as well in the integrated file system.

The path is as follows where <VE Log Folder> is equal to /qibm/userdata/VE2/logs:

- ▶ <VE Log Folder>/uninstallLog.txt// contains error messages
- ▶ <VE Log Folder>/uninstallDebug.txt// contains debug trace
- ▶ <VE Log Folder>/uninstallDebug.txt//

To display these files in iSeries Navigator or the 5250 emulation window, refer to 5.4.1, "Steps to display install log files using iSeries Navigator" on page 136 and 5.4.2, "Display install log files using CL command line" on page 137.

5.6 IBM Virtualization Engine Base Support

On the following Web site, you can find critical fixes for IBM Virtualization Engine by product name with a fix period:

<http://techsupport.services.ibm.com/server/VirtualizationEngine/home2.html>

Keep in mind if you have not had IBM Virtualization Engine installed at all before, the only possible fixes needed are only for the IBM Virtualization Engine Base Support.

Follow the instructions on the Web site to get these fixes installed or go to the infocenter for documentation on this:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r2/index.jsp?topic=/veicinfo/eicarfixparent.htm>

5.6.1 IBM Director server update

The recommended fixes for IBM Director Server 510 can be downloaded from the following Web site:

<http://www.ibm.com/pc/support/site.wss/document.do?lnodocid=SERV-DIRECT>

Click just under current version, scroll down until you see, for example, “IBM Director 5.10 Update 1 i5/OS Server Patch (requires Director 5.10 Server to be installed first)” and click the zip file next to it for download. The installation instructions are in a PDF file on the Web site:

ftp://ftp.software.ibm.com/pc/pccbbs/pc_servers_pdf/dir5.10.1_docs_relnotes.pdf

It is described in the section, Upgrading IBM Director Server on i5/OS, in Chapter 2, “Installation Upgrade” information of the above PDF. For the latest information, check this PDF.

Note: If you are not using IBM Director in a Virtualization Engine environment, you can use the Restore Licensed Program (RSTLICPGM) command to upgrade IBM Director Server to Version 5.10 Update 1.

Follow these steps (abstracted from the dir5.10.1_docs_relnotes.pdf) to install the update:

1. Extract the contents of the dir5.10.1_server_patch_i5os.zip file to a local directory. This archive contains the SAVDR100MM.sav file.
2. Use **FTP** to transfer SAVDR100MM.sav into a save file on the i5/OS system on which you want to upgrade IBM Director Server as follows:
 - a. Open a DOS prompt **start** → **Run** → **cmd**, or clicking on a shortcut for DOS prompt.
 - b. Type **ftp <your iSeries systemname or IP Address>** and press Enter.
 - c. A user ID is requested to log on to the iSeries FTP server, type, for instance, **QSECOFR** and press Enter.
 - d. Then the password is requested, type the password for the user at step 2c.
 - e. A message should return that the user is logged on. Type **cd /qsys.lib/qgp1.lib** and press Enter. The path on the iSeries is chosen this way and the name format being 1.
 - f. Type the local directory where you extracted the update file from the downloaded zip file for instance **lcd c:**.
 - g. Type **bin** for binary transfer.
 - h. Type the put command with the file from the zip you downloaded, for instance **put savdr100mm.sav savdr100mm.savf**.
 - i. Type **quit** to exit FTP and proceed with step 3.

```

C:\Documents and Settings\Administrator>ftp demo
Connected to Demo.
220-QTCP at p64321.rchland.ibm.com.
220 Connection will close if idle more than 5 minutes.
User (Demo:(none)): qsecofr
331 Enter password.
Password:
230 QSECOFR logged on.
ftp> cd /qsys.lib/qgpl.lib
250-NAMEFMT set to 1.
250 "/QSYS.LIB/QGPL.LIB" is current library.
ftp> lcd c:\
Local directory now C:\.
ftp> bin
200 Representation type is binary IMAGE.
ftp> put savdr100mm.sav savdr100mm.savf
200 PORT subcommand request successful.
150 Sending file to member SAUDR100MM in file SAUDR100MM in library QGPL.
226 File transfer completed successfully.
ftp: 147902304 bytes sent in 33.42Seconds 4425.83Kbytes/sec.
ftp> quit
221 QUIT subcommand received.

```

Figure 5-41 FTP example

3. Use RSTLICPGM to install the product. In the following example, SAVDR100MM is the name of the save file to which SAVDR100MM.sav was transferred:

RSTLICPGM LICPGM(5722DR1) DEV(*SAVF) SAVF(QGPL/SAVDR100MM)

Restore Licensed Program (RSTLICPGM)		
Type choices, press Enter.		
Product	> 5722DR1	Character value
Device	> *SAVF	Name, *SAVF
+ for more values		
Optional part to be restored . .	*BASE	*BASE, 1, 2, 3, 4, 5, 6, 7...
Type of object to be restored .	*ALL	*ALL, *PGM, *LNG
Language for licensed program .	*PRIMARY	Character value, *PRIMARY...
Output	*NONE	*NONE, *PRINT
Release	*FIRST	Character value, *FIRST
Replace release	*ONLY	Character value, *ONLY, *NO
Save file	> SAVDR100MM	Name
Library	> QGPL	Name, *LIBL, *CURLIB
More...		
F3=Exit	F4=Prompt	F5=Refresh
F10=Additional parameters	F12=Cancel	
F13=How to use this display	F24=More keys	

Figure 5-42 Restore Licensed Program (RSTLICPGM) window

4. Restart IBM Director Server as described 5.2.2, "Start IBM Director Server" on page 109.

5.7 iSCSI integrated server in multiple partitions

On each partition with an iSCSI connection to an xSeries or BladeCenter, IBM Directory server version 510 should be installed. If this is the case the default user ID and password should be changed on the service processor of the xSeries or BladeCenter preventing that the wrong IBM Director sever gets control on the xSeries' or BladeCenter's service processor before it is even connected to the network. Because this password is managed by i5/OS and

is used when your iSeries server starts a conversation with the xSeries or BladeCenter system's service processor. The service processor checks the password to ensure that the i5/OS configuration is authentic. Before the xSeries or BladeCenter is connected to the network, make sure the default user ID and password have been changed to a new user ID and password.

The IP Address used in the following two chapters are the default ones, if the IP Address has been change to another subnet, change the IP Address and/or subnet accordingly.

5.7.1 Create a new user on RSA II service processor

To create a new user ID and password on a service processor (RSA II) on an xSeries, follow the steps:

1. Set the IP address on your Windows workstation to something in the same subnet as the RSA II default IP address of 192.168.70.125, such as 192.168.70.124, and set the subnet mask to 255.255.255.0.
2. Open a Web browser. In the address or URL field, type the IP address 192.168.70.125 of the RSA II to which you want to connect. The Connect to window opens (Figure 5-43).

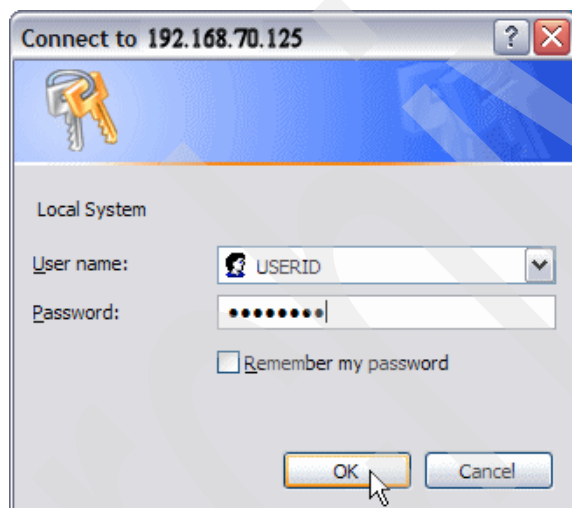


Figure 5-43 Connect to window of Service Processor

3. Type the User name and Password on the Connect to window. The RSA II has a default user name of USERID and password of PASSWORD (where 0 is a zero, not the letter O).
4. Select a **time-out value** on the next window and click **Continue**.
5. Select **Login Profiles** under **ASM Control** in the navigation pane on the left side of the window. The Login Profiles defined in the SP are shown in Figure 5-44 on page 142. Click the default user **USERID**.

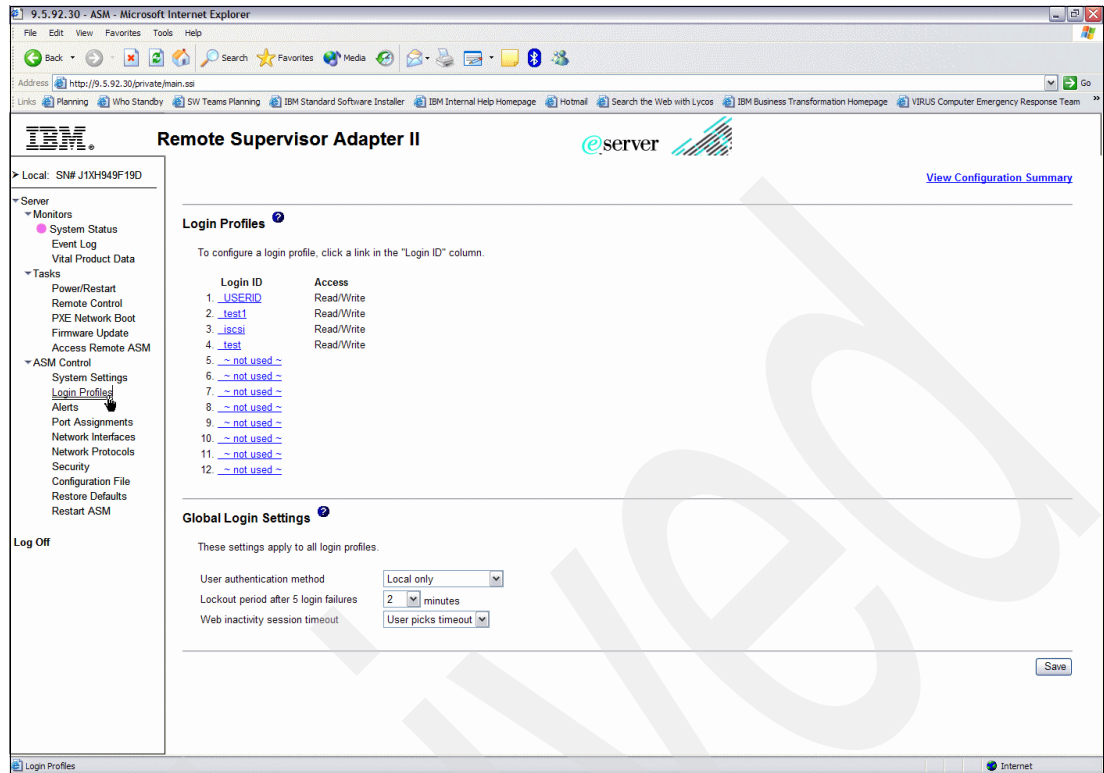


Figure 5-44 Login Profiles on RSAII

- The Login Profile window opens (Figure 5-45). On this window, change the **Login ID** and fill in the **Password** and **Confirm Password** field. Also make sure the Authority Level is set to **Supervisor**. Click **Save**.

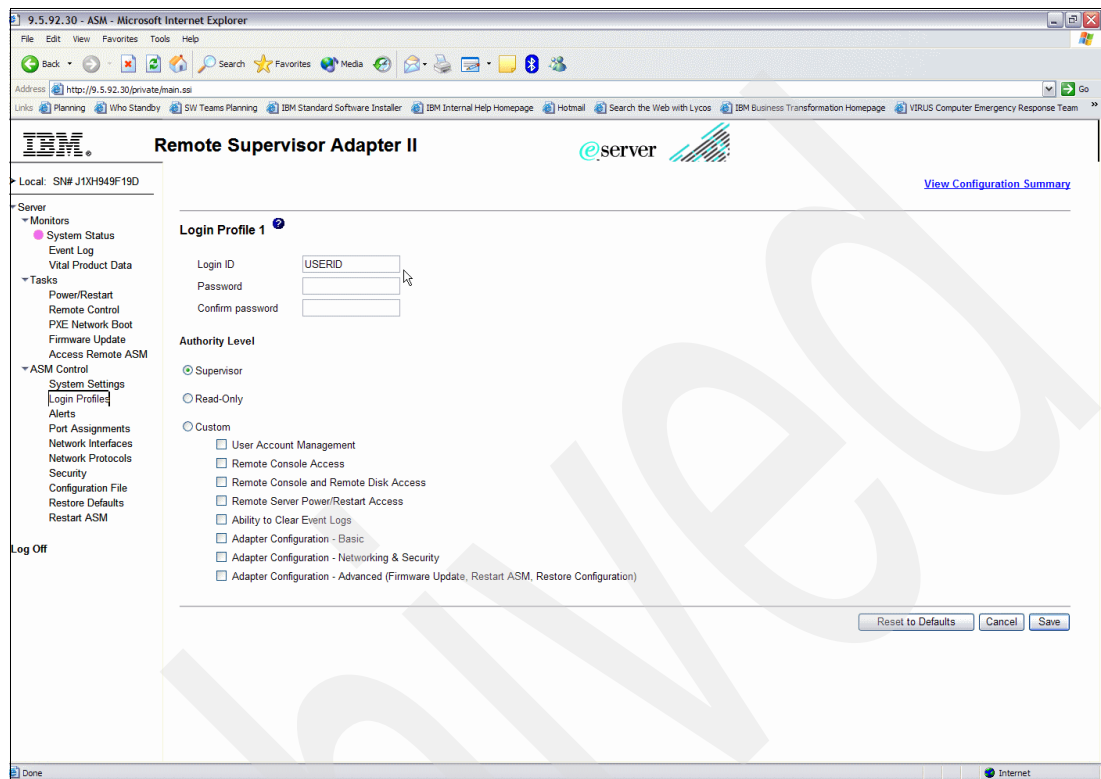


Figure 5-45 Login Profile properties on RSAII

- You are now returned on the Login Profiles window, Select **Log off** at the left bottom side in the left pane. This completes the change of the default Login Profile of the SP.
- The default user ID and password have now been changed. The new user ID and password are needed during the installation procedure in Chapter 4, "Installing the iSCSI integrated server" on page 53.

5.7.2 Create a new user on MM service processor

To create a new user ID and password on a Management Module (MM) of a BladeCenter, follow the steps:

- Set the IP address on your Windows workstation to something in the same subnet as the MM default IP address of 192.168.70.125, such as 192.168.70.124, and set the subnet mask to 255.255.255.0.
- Open a Web browser. In the address or URL field, type the IP address 192.168.70.125 of the MM to which you want to connect. The Connect to window (Figure 5-46 on page 144) opens.

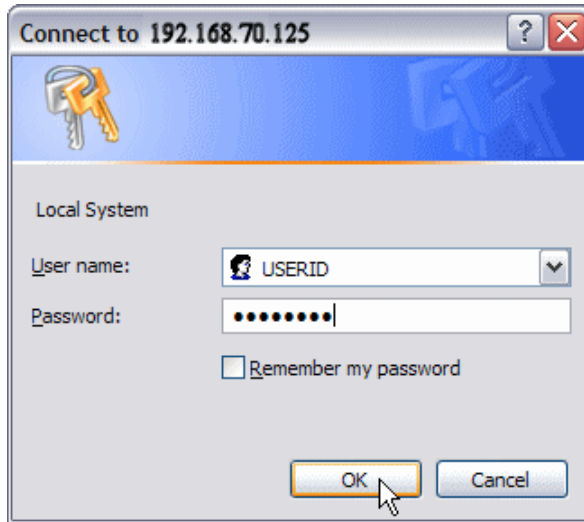


Figure 5-46 Connect to window of Service Processor

3. Type the **User name** and **Password** on the Connect to window. The MM has a default user name of USERID and password of PASSWORD (where 0 is a zero, not the letter O).
4. Select a **time-out value** on the next window and click **Continue**.
5. Select **Login Profiles** under **MM Control** in the navigation pane on the left side of the window. The Login Profiles defined in the MM are shown in Figure 5-47. Click the default user **USERID**.

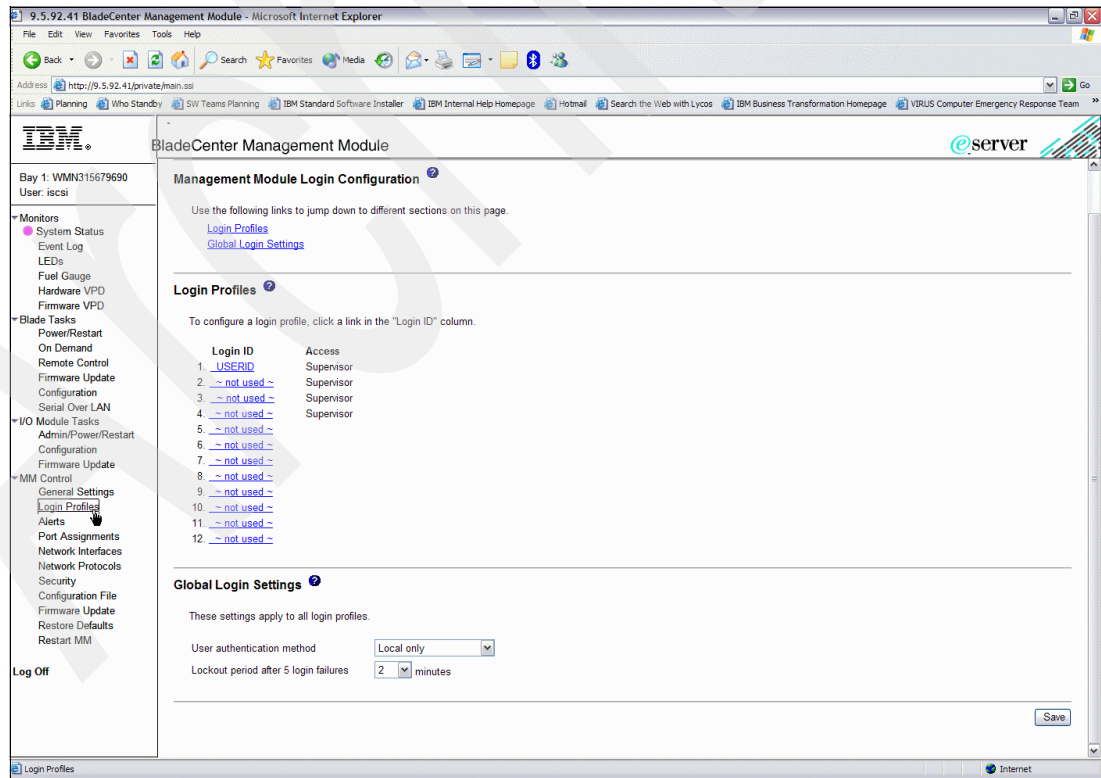


Figure 5-47 Login Profiles on MM

- The Login Profile window opens. On this window, change the **Login ID** and fill in the **Password** and **Confirm Password** field. Also make sure the **Authority Level** is set to Supervisor. Click **Save** in the lower right part of the window. You might have to scroll down to see this **Save** button.

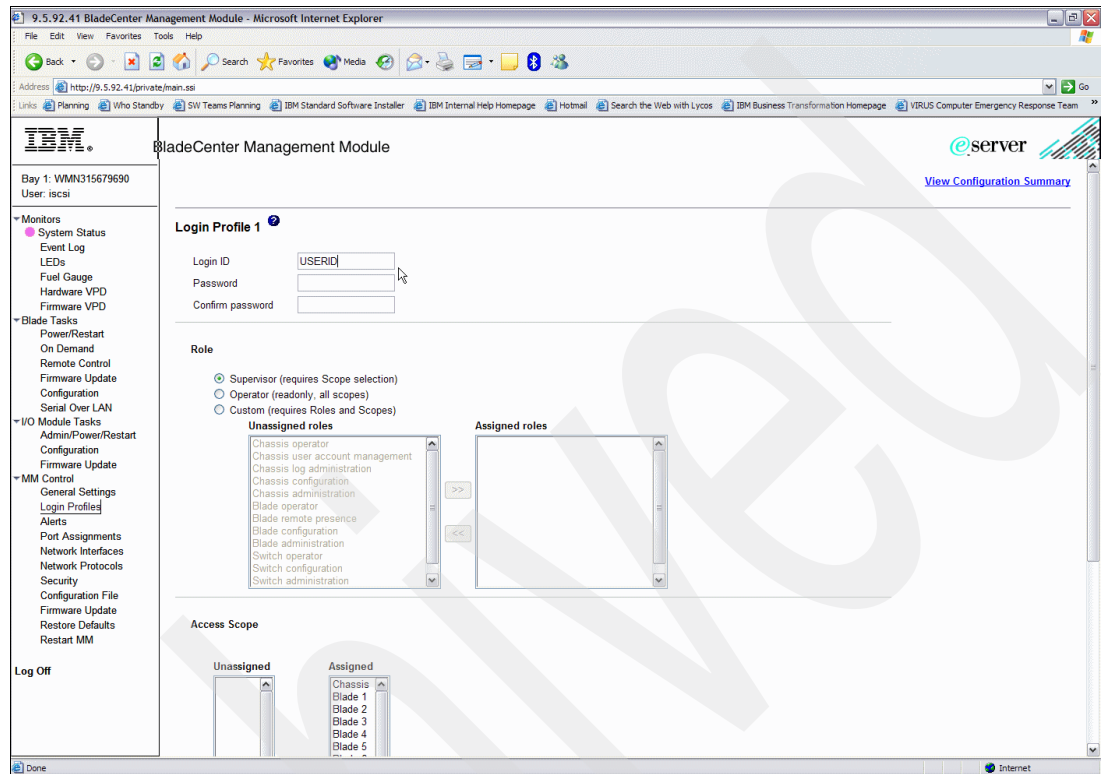


Figure 5-48 Logon Profile 1 window on MM

- You are now returned to the Login Profiles window. Select **Log off** at the lower left side in the left pane. This completes the change of the default Login Profile of the MM.
- The default user ID and password have now been changed. The new user ID and password are needed during the installation procedure in Chapter 4, "Installing the iSCSI integrated server" on page 53.

Note: You probably perform these steps once; still, there are a few options to manage the user ID and password, which are outlined in 6.5.1, "Initialize options for user ID and Password on SP/MM" on page 169.

5.8 QSHELL commands

QSHELL is an interface to run commands based on POSIX and X/Open standards.

There are a few commands, which you can run in the QSHELL interface (STRQSH). These commands reside in the integrated file system's directory `/qibm/proddata/director/bin`.

Note: We execute the commands directly, because it depends what the current directory is set for when executing `strqsh`. You could also change to the directory first by using the command `cd /qibm/proddata/director/bin` and pressing Enter. Then, type the command.

A few of the commands and their meanings are:

► **Twgstat** - Status of IBM Director Server:

- Type the command STRQSH on a command line and press Enter.
- Type the command `/qibm/proddata/director/bin/twgstat` and press Enter.
- The status returns to the window (Figure 5-49), in this case **Active**, and another status could be starting.

```
QSH Command Entry

$
> cd /qibm/proddata/director/bin
$
> twgstat
Active
$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top   F18=Bottom  F21=CL command entry
```

Figure 5-49 Status of IBM Director Server in QSHELL

Twgstart - start IBM Director Server:

- Type the command STRQSH on a command line and press Enter.
- Type the command `/qibm/proddata/director/bin/twgstart` and press Enter.

- The results are on the QSHELL (QSH) window (Figure 5-50) that the IBM Director Server starts.

```
QSH Command Entry

$
> /qibm/proddata/director/bin/twgstart
Service started.
$
> /qibm/proddata/director/bin/twgstat
Starting
$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 5-50 twgstart command in QSHELL

Note: You notice that even if it replied “Service started.”, it is not completely started, which is the same as how you start IBM Director Server using the command STRTCPSVR. That is why we did a **twgstat** to show you that the server is still starting. When the command **twgstat** returns **Active**, it is completely started.

- **Twgstop** - stop the IBM Director Server:
 - Type the command STRQSH on a command line and press Enter.
 - Type the command `/qibm/proddata/director/bin/twgstop` and press Enter.
 - The results are on the QSHELL window (Figure 5-51) that the IBM Director Server ends.

```
QSH Command Entry

$
> /qibm/proddata/director/bin/twgstop
Requesting service to end.
Waiting for service to end.
Service ended.
$

===>

F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear F17=Top  F18=Bottom  F21=CL command entry
```

Figure 5-51 twgstop command in QSHELL

- **Twgreset** - This command returns IBM Director Server to its installation default values and clears all tables in the database. Make sure that IBM Director Server is ended before executing **twgreset**!
 - Type the command STRQSH on a command line and press Enter.
 - Type the command `/qibm/proddata/director/bin/twgreset` and press Enter.
 - The results are on the QSHELL window (Figure 5-52) that the IBM Director Server will clear the tables. In the window, only part of all the tables are listed. When IBM Director Server is started again, it rebuilds these tables again. Depending on the size of the environment, it can take awhile to start.

```

QSH Command Entry

$
> /qibm/proddata/director/bin/twgreset
Table HMC_CEC_LPAR was successfully deleted
Table SMIS_SYSTEMS was successfully deleted
Table UMS_CONFIG was successfully deleted
Table TWG_IDE_ADAPTER was successfully deleted
Table TWG_PARALLEL_PORT was successfully deleted
Table TWG_DISK was successfully deleted
Table TWG_PROCESSOR was successfully deleted
Table TWG_VIDEO was successfully deleted
Table TWG_FIBRE_ADAPTER was successfully deleted
Table MP_VPD was successfully deleted
Table TWG_MANAGED_OBJECT was successfully deleted
Table TWG_LOGICAL_MEMORY was successfully deleted
Table ..... successfully deleted
DeleteTables returns 0
Successfully restored product to its original state.
$

==>

F3=Exit  F6=Print F9=Retrieve F12=Disconnect
F13=Clear F17=Top F18=Bottom F21=CL command entry

```

Figure 5-52 *twgreset* command in QSHELL

Note: The start of IBM Director Server takes longer after a reset due to the rebuild of the deleted tables.

Managing integrated iSCSI environments

This chapter describes the management tasks necessary to operate your iSCSI integrated server. It describes the features and highlights the unique way iSCSI integrated servers interact with iSeries. It describes how to manage your iSCSI integrated server using iSeries Navigator and CL command line, which include the new objects used. Within this chapter, there are references to a xSeries or a Blade, which resides in a BladeCenter.

Terminology: In this chapter, the reference *iSCSI server* applies equally to *iSCSI integrated server* or *iSCSI integrated Windows server*. The term *integrated Windows server*, or just *integrated server*, refers to an instance of Microsoft Windows Server 2003 on an iSCSI integrated Server, an xSeries, or IBM BladeCenter server attached to an iSeries server with an iSCSI Host Bus Adapter (HBA).

The xSeries and BladeCenter have their own Service Processor: for the xSeries it is called a *Service Processor* or *SP* and for the BladeCenter it is called the *Management Module* or *MM*.

The term *Blade* refers to IBM BladeCenter server in a BladeCenter chassis.

The term *Storage Space* also applies to Network Server Storage Space. This is i5/OS single level disk storage space allocated to an integrated server that is seen as a disk drive on the xSeries or BladeCenter.

6.1 Introduction to management concepts

With iSCSI, several new i5/OS objects and commands are introduced in release 540. Due to these new i5/OS objects involved, the management of an iSCSI server is different than it was before with the well known Integrated xSeries Server (IXS) and the Integrated xSeries Adapter (IXA) as described in Chapter 1, “Introduction to iSCSI integrated server support” on page 1 and Chapter 3, “Planning for iSCSI attached servers” on page 43. One new capability to be able to connect to a BladeCenter with several Blades. From a management point of view, there is not much difference between an xSeries server and a Blade.

The iSCSI server needs the new i5/OS objects and configuration information, because iSCSI attached servers are connected to the i5 system with a Ethernet network to identify and communicate with the xSeries or BladeCenter server on the network. Therefore, you need to be able to manage these objects.

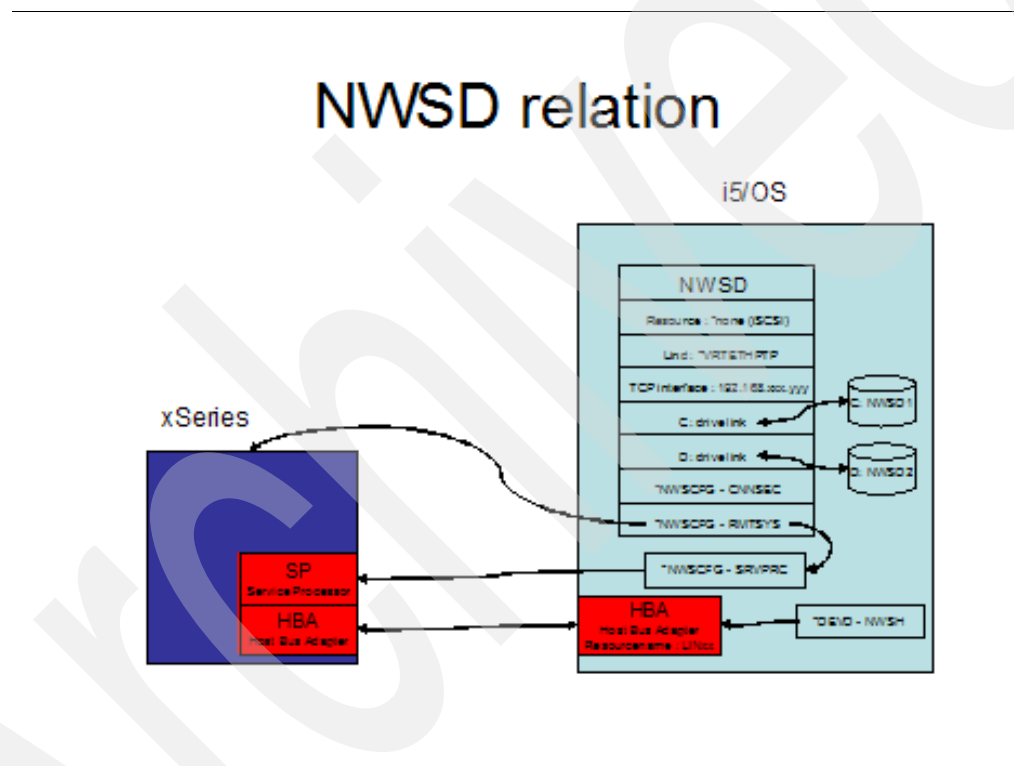


Figure 6-1 Relationship between NWSD in i5/OS and an xSeries

When an iSCSI integrated Windows server is installed on an i5 system, the following objects are created:

► **Network server host adapter**

Network server host adapter (DEVD subtype NWSH) objects are used to configure the i5 system target iSCSI host bus adapter (iSCSI HBA). An NWSH object must be started (varied on) in order for an integrated server to use the corresponding iSCSI HBA for storage or Virtual Ethernet data flows. Stopping (varying off) a NWSH object will make the corresponding iSCSI HBA unavailable to any integrated servers that have storage or Virtual Ethernet paths defined to use it (see Figure 6-1 on page 150).

► **Remote system configuration**

Remote system server configuration (NWSCFG subtype RMTSYS) objects are used to configure attributes of an iSCSI attached remote xSeries server or Blade server in an IBM BladeCenter chassis that the integrated server will run on. It also defines how the remote system boots and communicates with the i5 system.

► **Service Processor configuration**

Service processor configuration (NWSCFG subtype SRVPRC) objects are used to configure attributes of the Service Processor of each iSCSI attached remote xSeries or of the Management Module of each iSCSI attached IBM BladeCenter server. The service processor configuration defines attributes that are used to discover and connect to the Service Processor or Management Module on the network. Remote system configuration objects contain a reference to the corresponding service processor configuration object that is used to control the remote system hardware.

Note: A service processor configuration is not needed for each Blade in a IBM BladeCenter chassis. Just one service processor configuration is needed for the IBM BladeCenter chassis.

► **Connection security configuration**

Connection security configuration (NWSCFG subtype CNNSEC) objects need to exist. At the time of this writing, it just needs to exist and will be used in the future to secure the storage and Virtual Ethernet data flows over the iSCSI network between the i5 system and xSeries server or Blade servers.

► **Network server description**

The network server description (NWSD) is still the key i5/OS configuration object for iSCSI integrated servers. The NWSD object is used to tie together the other i5/OS objects that relate to an iSCSI integrated server. With iSCSI, there is a difference on some points:

- It contains a reference to a remote system configuration object instead of an iSeries hardware resource name.
- Unlike an IXA attached server, which uses one IXA card in the i5 system to manage all data flows, on an iSCSI attached server both the i5 system and xSeries can have multiple iSCSI host bus adapters (HBAs). This allows multiple SCSI and Virtual Ethernet data paths between the i5 system and xSeries or IBM BladeCenter, which can provide greater bandwidth and connection redundancy.
- You can define one or more storage paths. These storage paths reference the NWSH device description objects that are associated with the iSCSI HBAs that are used by the iSCSI integrated server. You can choose which storage path is used for each virtual disk drive.

- You can define one or more Virtual Ethernet paths. These Virtual Ethernet paths also reference the NWSH device description that is used by the iSCSI integrated server. You can choose which NWSH is used for each Virtual Ethernet port that the integrated server uses.
- The iSCSI attached xSeries or IBM BladeCenter server hardware is controlled by i5/OS.
 - An iSCSI attached server is started and shut down by varying on or off the NWSD for that server.

Note: Due to the iSCSI concepts, it is possible to use **start** → **shutdown** from the Windows desktop console without any problem, you can even start the iSCSI connected server by just pressing the power button if the **start** → **shutdown** from the Windows desktop console is done.

- For an iSCSI attached xSeries or IBM BladeCenter server, i5/OS communicates over an Ethernet network with the Service Processor (SP) for the xSeries system or the IBM BladeCenter Management Module (MM) for an IBM BladeCenter system to perform the start and shutdown tasks.

► Network server storage spaces

Because xSeries or IBM BladeCenter servers connected to an i5 system using iSCSI do not have disk of their own, they use i5/OS single level disk storage for storing the system drive, the source drive, client data, and sharing network files. i5/OS disk storage allocated to an integrated server is called *network server storage space* or *storage space* for short. The integrated server equivalent of installing a new hard drive in a PC server is to create a network server storage space in i5/OS and link it to an integrated server. Network server storage spaces can reside in the system ASP, a user ASP, or an Independent ASP (IASP), and appear as objects in the /QFPNWSSTG IFS directory.

The install command (INSWNTSVR) creates only two drives automatically, but additional storage spaces can added, which are more likely used for applications and data:

- System drive (the C: drive) contains the Windows Server operating system, in an iSCSI integrated server, this is Windows Server 2003.
- Installation drive (the D: drive) contains a copy of the Windows Server installation media as well as the portion of the i5/OS Integrated Server Support (5722-SS1 option 29) code that runs on the Windows Server. The installation drive is used during the Windows Installation process and is also used every time the server is started to pass configuration information from i5/OS to the server.
- Additional user-defined drives are typically used for server applications and data.

Virtual disk drives can vary in size from 1 MB to 1024000 MB (1TB) each. Up to 64 virtual disks can be linked to a server, depending on the server configuration, so the storage capacity of an integrated server can be more than many terabytes.

► Virtual Ethernet line descriptions

i5/OS needs a way to communicate with its integrated Windows servers. The communication takes place over a point-to-point Virtual Ethernet network. The Virtual Ethernet point-to-point line description and port number value, which is *VRTETHPTP on the i5/OS, is created during the installation command (INSWNTSVR). Also, on the Windows side, a Virtual Ethernet point-to-point is created as well. So there is a virtual point-to-point connection (without cables and adapters) between the i5/OS system and the integrated server. Virtual Ethernet configuration is discussed in Chapter 8, “Virtual Ethernet LAN” on page 305.

- **TCP/IP interface for Point-to-Point Virtual Ethernet LAN line**

For an iSCSI integrated server, these addresses take the form of 192.168.xxx.yyy, where xxx ranges from 100 to 254 and results in a unique class C network. For example, the i5/OS side point-to-point network will be given the IP address 192.168.100.1, and the Windows side has 192.168.100.2. As you define multiple iSCSI integrated servers, xxx increments by 1, so, in our example, the next iSCSI server will be given the IP address 192.168.101.1 on the i5/OS side and 192.168.101.2 on the Windows side.

Note: You can change these IP addresses on the INSWNTSVR command to prevent TCP/IP address conflicts with other hosts on the system.

- **Windows Services**

After an installation of an iSCSI integrated server, five services are installed on Windows as part of the integration code. These Windows Services have their own function.

The Windows Services and their function are:

- **iSeries Manager**

Manages integrated server startup and shutdown operations for *non-iSCSI* integrated servers (IXS and IXA).

- **iSeries Remote Command**

Enables processing Windows commands from i5/OS.

- **iSeries Shutdown Manager**

Enables system shutdown from i5/OS over the iSCSI LAN Network. This service does a query every 10 seconds if a request was done for a shutdown. If the query response is yes, a shutdown is issued.

Note: Never stop this iSeries Shutdown Manager because the xSeries or BladeCenter might not respond to a shutdown request from i5/OS.

- **iSeries User Administration 6.15**

Supports user enrollment, event log, disk, and statistic service request from i5/OS.

- **iSeries Virtual Ethernet Manager**

Manages the connection status for iSCSI-based Virtual Ethernet network adapters.

Note: If this iSeries Virtual Ethernet Manager is stopped, the xSeries or BladeCenter do not respond to any Virtual Ethernet link state changes.

To verify the status of the services, click **Start → Programs → Administrative Tools → Services**. The service should have a status of Started. See Figure 6-2 on page 154.

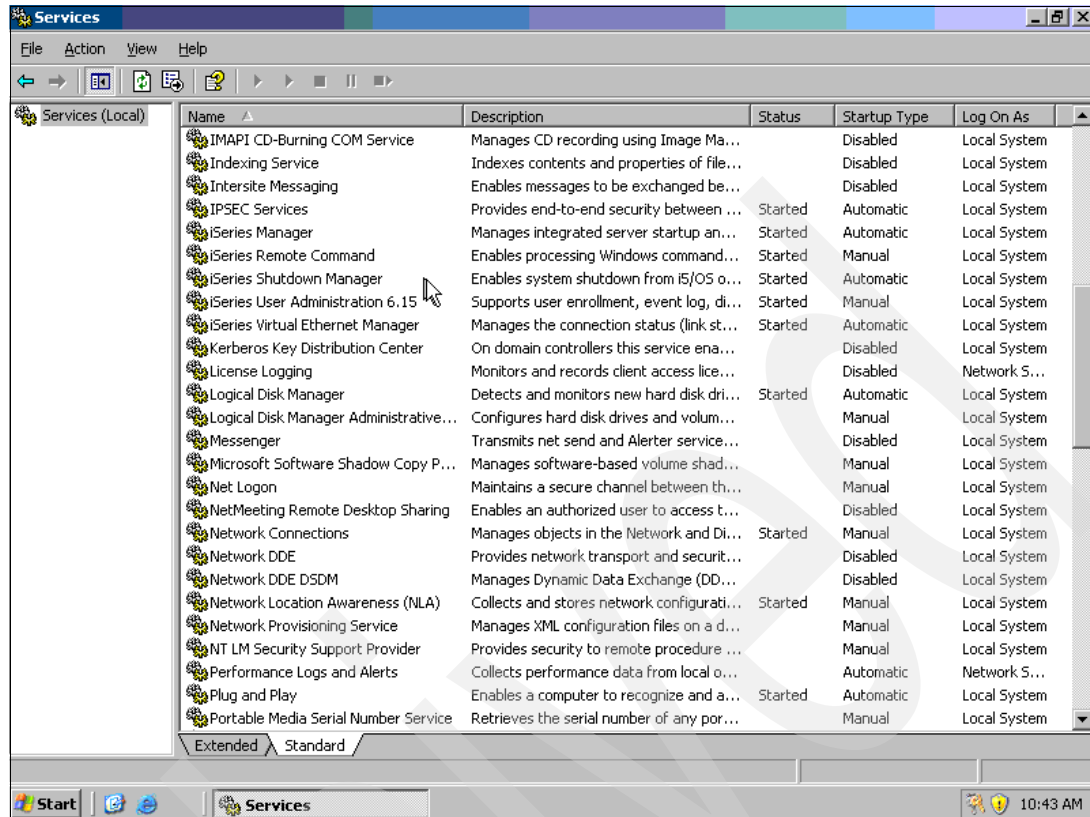


Figure 6-2 Integrated Server Windows Services

6.2 Choose your two possible interfaces

The iSeries has two interfaces to manage your iSCSI connections:

- Command Line interface (also known as *green screen*)

The commands used to manage iSCSI connections are listed at each function mentioned using iSeries Navigator.

- iSeries Navigator because it is a graphical user interface (GUI)

Before you can use iSeries Navigator, you must install it. You can install it from the iSeries Access CD-ROM, or direct from the iSeries itself, providing that the 5722-XE1 licensed program product is installed.

Note: You do not need to have 5722-XE1 installed in order to use iSeries Navigator on your Windows PC. For more information about installing iSeries Navigator, refer to *Managing OS/400 with Operations Navigator V5R1 Volume 1: Overview and More*, SG24-6226. You can download this document from the IBM Redbooks Web site:

<http://www.redbooks.ibm.com/>

Note: iSeries Navigator requires that 5722-SS1, option 12 (OS/400 - Host Servers) is installed on your iSeries. This option should be installed by default, but you can check by using the Go Licensed Programs command (GO LICPGM) and selecting option 10.

After you install iSeries Access on Windows, you must download and install the latest service pack from this Web site:

<http://www.ibm.com/servers/eserver/iseries/access/casp.htm>

In iSeries Navigator release 540, the Windows administration has moved to a new topic right under YourSystem. The name is also changed from “Windows Administration” to “Integrated Server Administration”. See Figure 6-3.

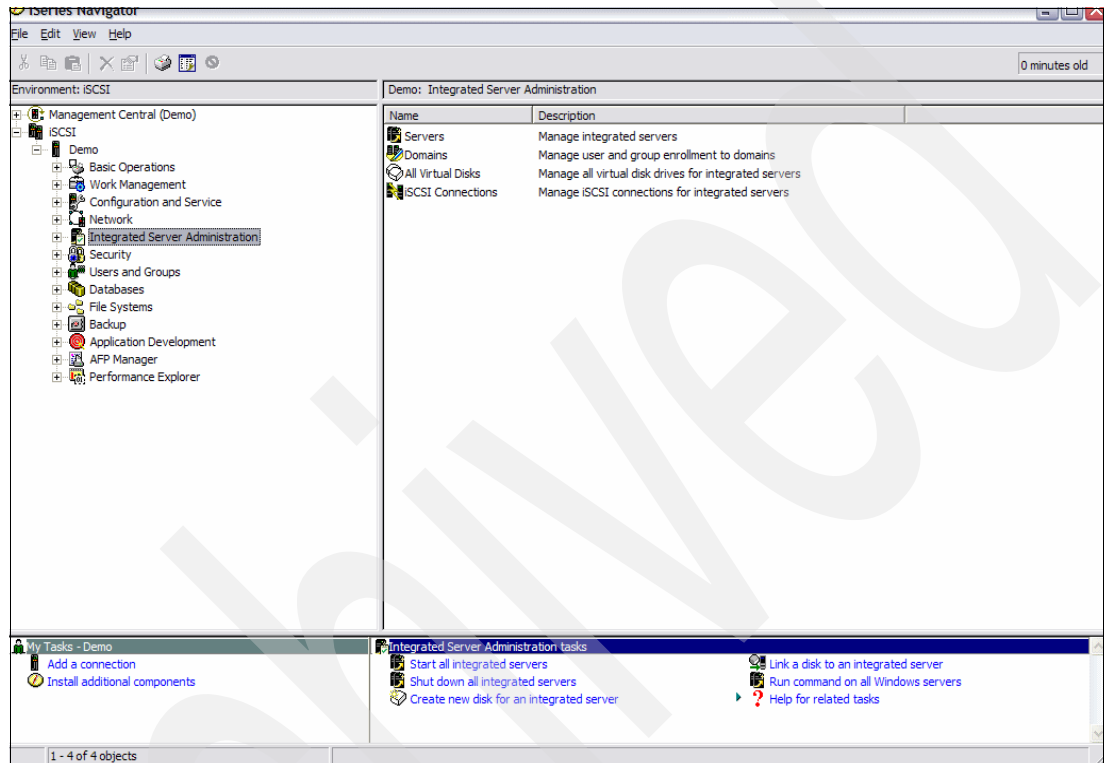


Figure 6-3 Integrated Server Administration

6.3 Working with iSCSI integrated server

Due to the new technology for iSCSI to connect xSeries and BladeCenter, there are new objects involved. So managing integrated servers has been expanded to work with these new objects. These new objects are described in 6.1, “Introduction to management concepts” on page 150.

To administer the integrated server administration in iSeries Navigator, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration**.
2. You see the following list:

Servers	To manage your created integrated servers, these could be an Integrated xSeries Server (IXS), Integrated xSeries Adapter (IXA), or an iSCSI Integrated Server.
Domains	To manage your domains, in which these Servers are, for enrolling users and groups.
All Virtual Disks	To manage all the virtual disk you created, which can be linked or not.

iSCSI Connection To manage the new objects (NWSH, SRVPRC, RMTSYS, and CNNSEC) used for iSCSI Integrated Server.

To use the integrated server administration using a command line interface compared to the previously mentioned iSeries Navigator administration, follow the steps:

1. Open a 5250 emulation window.
2. Use the following CL commands to manage the Integrated Server:

Servers	WRKNWSD → Work with Network Servers display appears.
Domains	WRKNWSEN → Work with NWS User Enrollment display appears.
All Virtual Disks	WRKNWSSTG → Work with NWS Storage Spaces display appears.
iSCSI Connections	WRKNWSCFG → Work with NWS Configuration display appears.

6.4 Manage network server host adapters

The network server host adapters (NWSH) are used to configure the iSeries target iSCSI host bus adapter (HBA).

6.4.1 Manage NWSH objects using iSeries Navigator

To manage NWSH objects using iSeries Navigator:

- **Create a network server host adapter object using iSeries Navigator.**

To create a network server host adapter object (NWSH), follow these steps:

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Right-click **Local Host Adapter** and select **New Network Server Host Adapter**.
- c. The New Network Server Host Adapter window opens. On the General tab, enter the following (Figure 6-4 on page 157):
 - i. Device Name.
 - ii. Device Description.
 - iii. Select the Hardware Resource using the pull-down list.
 - iv. If required, select **Online at IPL**.
 - v. Select the **Object authority** and select the Communications tab.

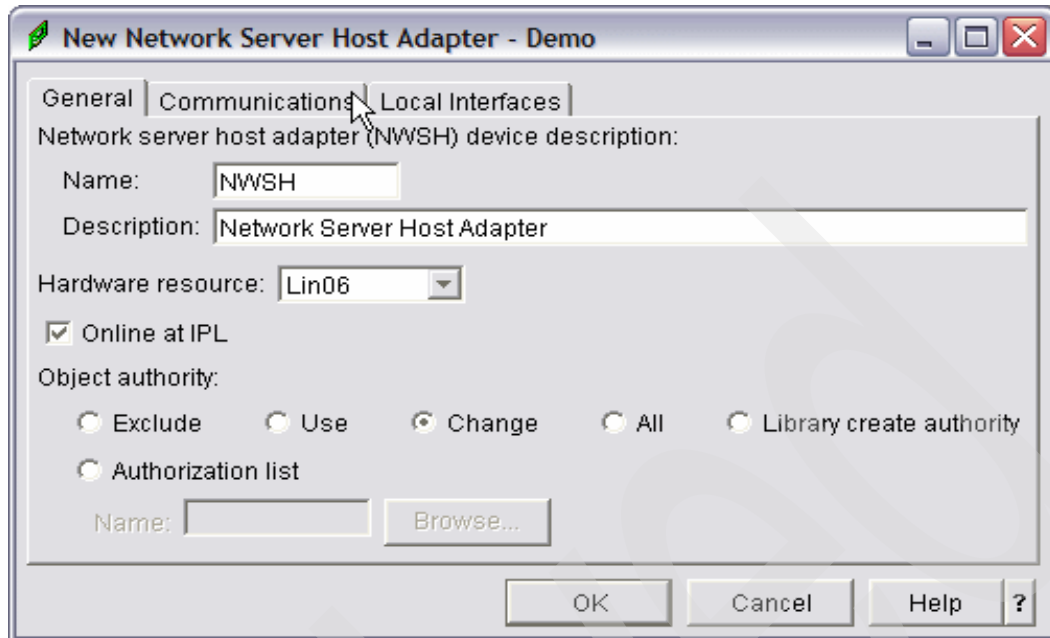


Figure 6-4 General tab: New Network Server Host Adapter window

- d. On the Communications tab, enter the following (Figure 6-5):
 - i. Enter an existing **Message queue** and **Library**, or leave the default System operator, you could also select **Browse** to search for an Message queue after you enter the message queue and library or use an option from the pull-down menu.
 - ii. Select the **Communications recovery limits**. Use system values refers to the system value QCMNRCYLMT or specify the values by selecting Use the following values. The default is shown.
 - iii. Click Local Interfaces tab.

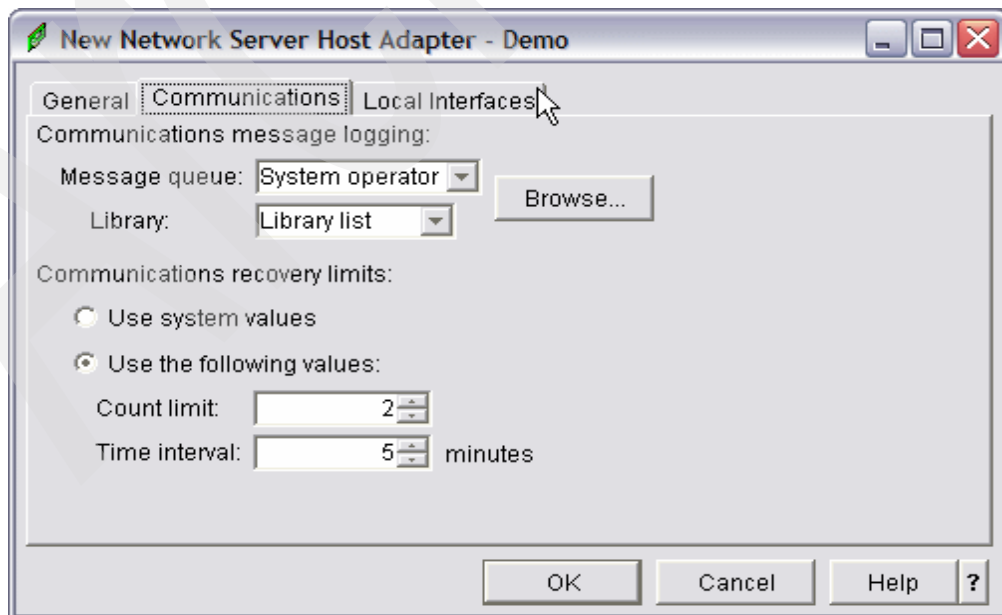


Figure 6-5 Communications tab: New Network Server Host Adapter window

- e. On the Local Interfaces tab, enter the following (Figure 6-6):
 - i. Enter the Subnet mask for both IP Addresses.
 - ii. Enter the **Local SCSI interface Internet address** and **Gateway address**. You can change the TCP port as well if needed, otherwise leave the default value.
 - iii. Enter the **Local LAN interface Internet address** and **Gateway address**. You can change the TCP port as well if needed, otherwise leave the default value. Click **OK**. The Network Server Host Adapter is created.

Note: At the time of writing this book, the Gateway address is required but is not used.

The screenshot shows a Windows-style dialog box titled "New Network Server Host Adapter - Demo". It has three tabs: "General", "Communications", and "Local Interfaces". The "Local Interfaces" tab is selected. The dialog contains the following fields and values:

- Subnet mask: 255.255.0.0
- Local SCSI interface:
 - Internet address: 128.168.201.1
 - Gateway address: 128.168.200.1
 - TCP port: 3260 (with a list of other possible values: 3260, 860, 1024-65535)
- Local LAN interface:
 - Internet address: 128.168.202.1
 - Gateway address: 128.168.200.1
 - Base virtual Ethernet port: 8801 (with a list of other possible values: 8801, 1024-65471)

At the bottom right, there are three buttons: "OK", "Cancel", and "Help". A mouse cursor is pointing at the "OK" button.

Figure 6-6 Local interface tab: New Network Server Host Adapter

► **Create a network server host adapter object based on another one using iSeries Navigator.**

To create a NWSH based on another one, follow the steps (Figure 6-7 on page 159):

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Click **Local Host Adapter** and right-click the **Network Server Host Adapter** at the right pane, which you want to be copied from and select **New Based On**.
- c. Enter a new device name and change any other attribute as described in Create a network server host adapter object using iSeries Navigator and click **OK**.

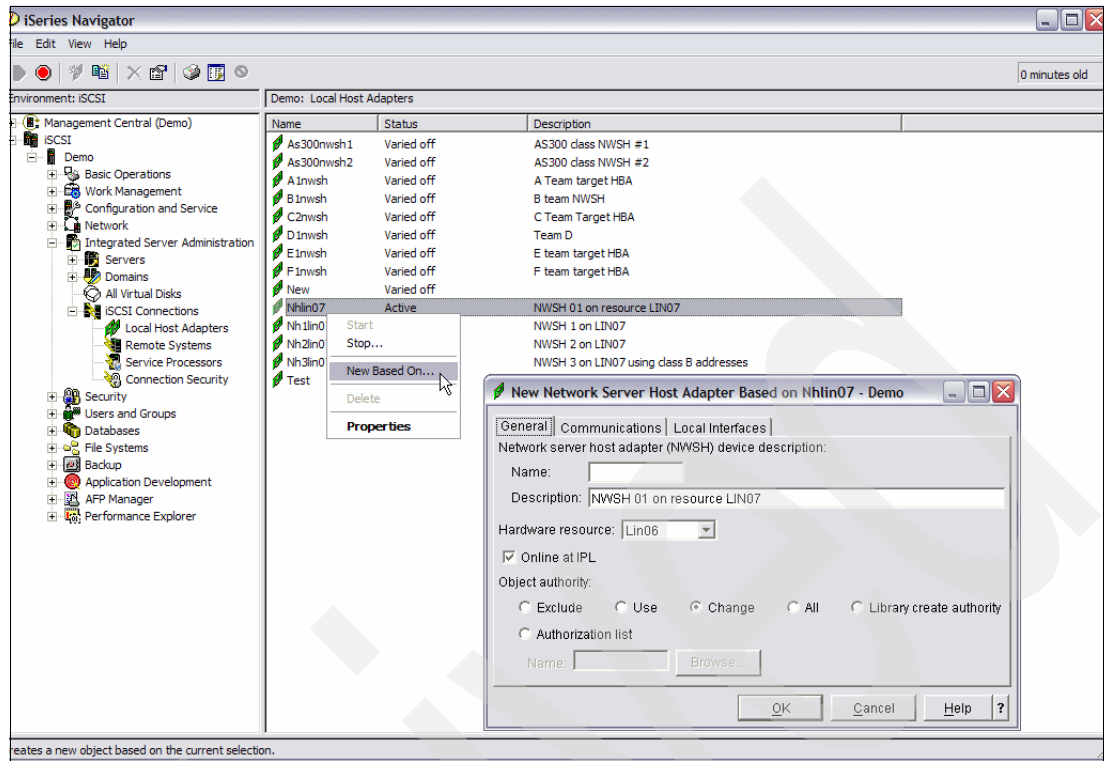


Figure 6-7 New Based on option on a Local Host Adapter

► **Display/Change network server host adapter using iSeries Navigator.**

To display and change the NWSH object, you follow the same steps. It depends if the NWSH is active or inactive, if you want to change attributes. With the NWSH active, you can change the following attributes:

On the General tab:

- Description
- Online at IPL

On the Communications tab:

- Communications message logging message queue and library
- Communications recovery limits

On the other two tabs, Local Interfaces and Resource Usage, nothing can be changed.

With the NWSH inactive, you can change almost all attributes except:

- Name on the General tab
- the Resource Usage tab

To display or change a network server host adapter, follow the steps:

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Click **Local Host Adapter** and right-click the **Network Server Host Adapter** at the right pane, which you want to display/change, and select **Properties** (Figure 6-8 on page 160).

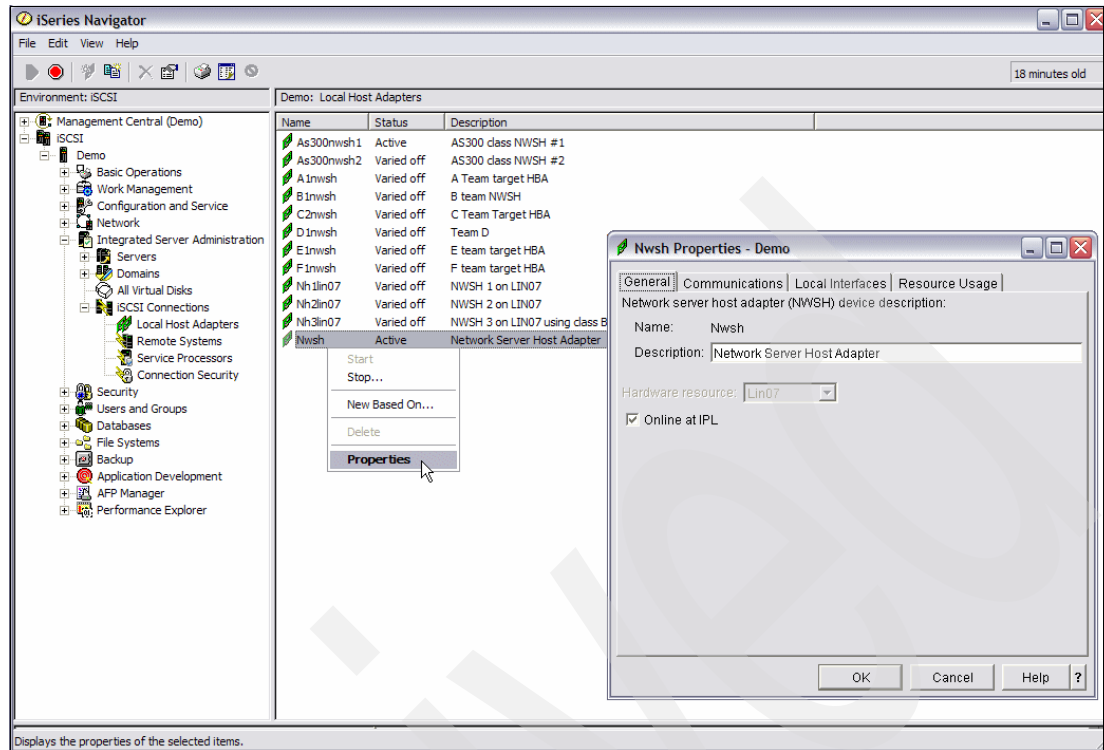


Figure 6-8 Properties NWSH object

► **Start a network server host adapter using iSeries Navigator.**

To start a network server host adapter, follow the steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Local Host Adapter** and right-click the **Network Server Host Adapter** at the right pane, which you want to start, and select **Start**. The status will flow from Varied off to Vary on Pending to Active (Figure 6-9 on page 161).

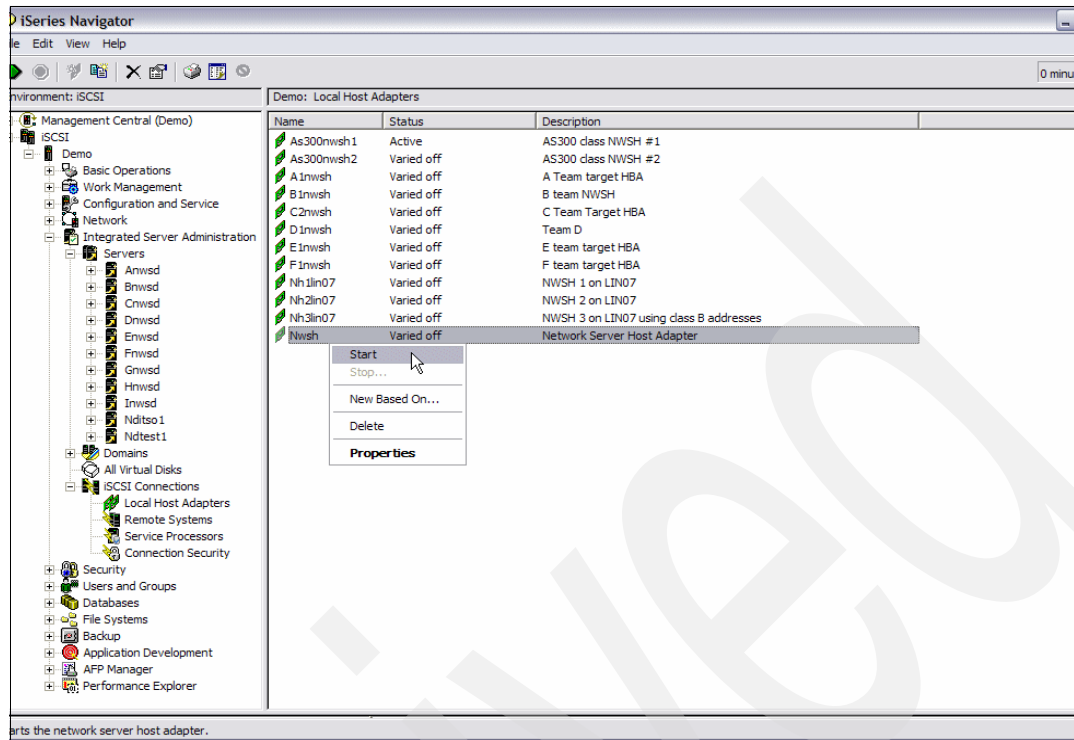


Figure 6-9 Start Local Host Adapter

► **Stop a network server host adapter using iSeries Navigator.**

To stop a network server host adapter (NWSH) object, first make sure there is no active iSCSI integrated server active using this NWSH. Because if the NWSD is active and the NWSH is stopped, the servers using this NWSH can fail if critical storage resources can no longer be accessed, including Virtual Ethernet paths which might use the same NWSH. If you, by accident, stop a NWSH and it has an active NWSD, a window to confirm the stop opens, showing a column “Active NWSDs” with “Yes”. At this time, you can select cancel. If you select “Stop”, you receive another window to confirm the action, stating if you proceed, the active servers will fail. An example is shown in Figure 6-10 on page 162. For stopping the iSCSI integrated server, see 6.9, “Stopping iSCSI integrated server” on page 231.

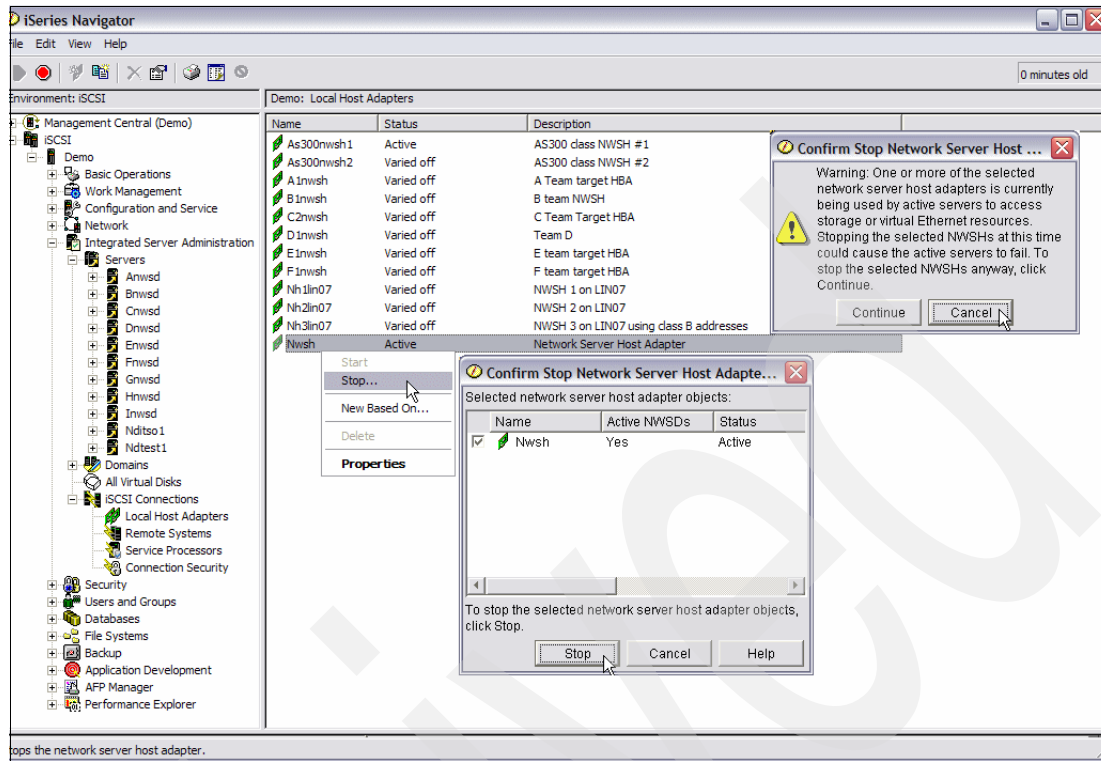


Figure 6-10 Stop a NWSH confirm messages

So here are the steps to stop a Local Host Adapter:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Local Host Adapter** and right-click the **Network Server Host Adapter** at the right pane, which you want to stop, and select **Stop**, the status will flow from Active to Vary off pending to Varied off.
- On the Confirm Stop Network Server Host Adapter window telling you if a NWSD is active or not, select **Stop**. This window always opens when you stop a NWSH. If the second Confirm Stop window opens, warning you that a NWSD is active, and that you should select **Cancel** and stop the active NWSDs first.

► **Delete a network server host adapter using iSeries Navigator.**

Here are the steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Local Host Adapter** and right-click the **Network Server Host Adapter** at the right pane, which you want to delete, and select **Delete**. Delete can only be selected when the NWSH is varied off.
- On the Confirm Delete Network Server Host Adapter window, select **Delete**. See Figure 6-11 on page 163.

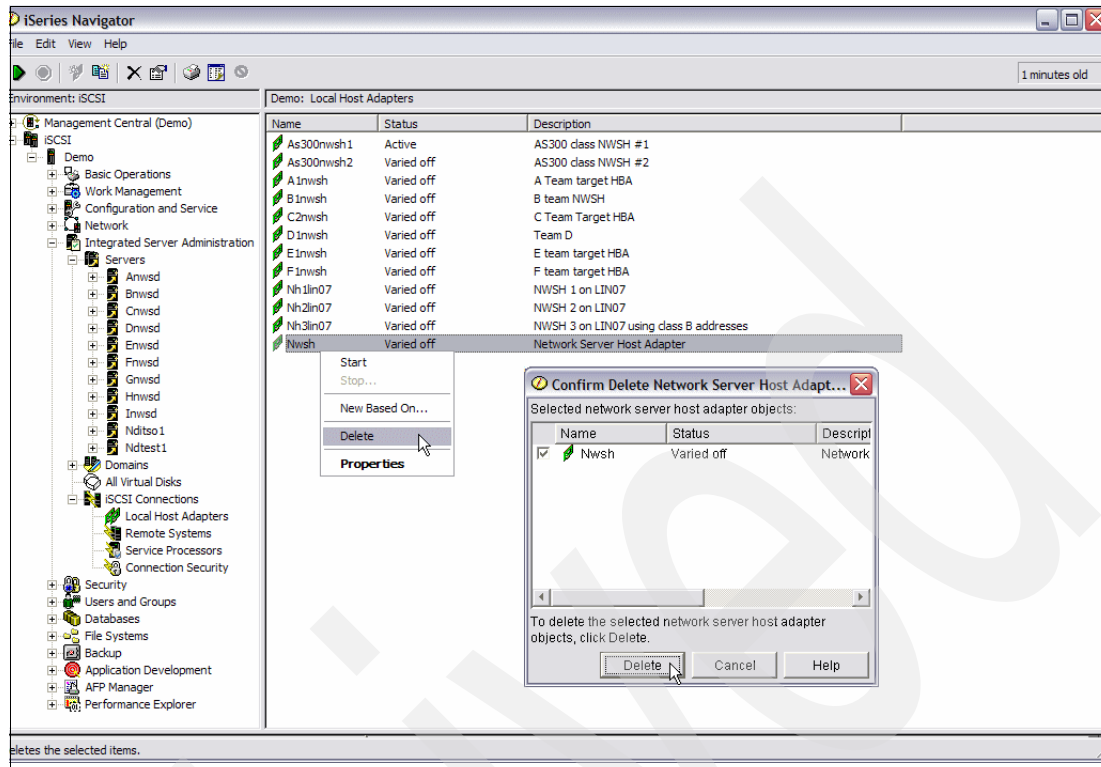


Figure 6-11 Delete Network Server Host Adapter

6.4.2 Manage NWSH objects using CL commands

To create a network server host adapter object using CL commands:

► Create a network server host adapter object using CL commands.

To create a network server host adapter object (NWSH), follow these steps:

- On a 5250 command line, type `CRTDEVNWSH` and press Enter or F4-prompt.
- The Create Device Desc (NWSH) display appears. See Figure 6-12 on page 164 and Figure 6-13 on page 164.
- Type a Device description (DEVD) name and Resource name (RSRCNAME) and press Enter for more parameters.
- Type the Subnet mask (LCLIFC) for both IP addresses.
- Type the Local SCSI interface Internet address (LCLIFC) and Gateway address (LCLIFC).
- Type the Local LAN interface Internet address (LCLIFC) and Gateway address (LCLIFC).
- If required, change the **Online at IPL** (ONLINE) parameter to *YES.
- Press page-down and change the **Message queue** and **Library** (MSGQ) if required.
- Change the **Recovery Limits: Count limit** and **Time interval** (CMNRCYLMT) if required.
- Type a Description for **Text 'description'** (TEXT) and press Enter. You will see a message at the bottom of the display confirming the device is created.
- To change the **Authority** (AUT), you have to press F10-Additional parameters.

The only option for the parameter **Local interface (LCLIFC): Port Speed** and **Duplex** is ***AUTO**.

We skipped the following parameters because they can be left at the default value, but can be changed if needed:

- **Local SCSI interface (LCLIFC): SCSI TCP port**
- **Local LAN interface (LCLIFC): Virtual Ethernet base UDP port**

```

Create Device Desc (NWSH) (CRTDEVNWSH)

Type choices, press Enter.

Device description . . . . . DEVD          > NWSH
Resource name . . . . . RSRNAME          > LIN07
Local interface:      LCLIFC
  Subnet mask . . . . .                255.255.0.0
  Port speed . . . . .                *AUTO
  Duplex . . . . .                    *AUTO
Local SCSI interface:
  Internet address . . . . .          128.168.201.1
  Gateway address . . . . .          128.168.200.1
  SCSI TCP port . . . . .            3260
Local LAN interface:
  Internet address . . . . .          128.168.202.1
  Gateway address . . . . .          128.168.200.1
  Virtual Ethernet base UDP port      8801
Online at IPL . . . . . ONLINE        *YES

More...
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 6-12 First page of the command CRTDEVNWSH

```

Create Device Desc (NWSH) (CRTDEVNWSH)

Type choices, press Enter.

Message queue . . . . . MSGQ          *SYSOPR
Library . . . . .
Recovery limits:      CMNRCYLMT
  Count limit . . . . .                2
  Time interval . . . . .              5
Text 'description' . . . . . TEXT      Network Server Host Adapter

Additional Parameters

Authority . . . . . *CHANGE          Name, *CHANGE, *ALL, *USE...

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 6-13 Second page of the command CRTDEVNWSH

► **Create a network server host adapter object based on another one using CL commands.**

To create a NWSH based on another one, follow the steps:

- On a 5250 command line, type **WRKDEVD *NWSH** and press Enter.
- The Work with Device Descriptions display appears (Figure 6-14).
- Scroll down to the NWSH you want copied and type 3 (Copy) on the option column (Opt) in front of the NWSH and press Enter.

Work with Device Descriptions				System: ITCDEM02
Position to		Starting characters		
Type options, press Enter.				
2=Change		3=Copy	4=Delete	5=Display
8=Work with status		9=Retrieve source		
Opt	Device	Type	Text	
3	NWSH	*NWSH	Network Server Host Adapter	
				Bottom
Parameters or command				
===>				
F3=Exit	F4=Prompt	F5=Refresh	F6=Create	F9=Retrieve
F14=Work with status		F12=Cancel		

Figure 6-14 WRKDEVD display option 3

► **Display network server host adapter using a CL command.**

To display a network server host adapter, follow the steps:

- On a 5250 command line, type **DSPDEVD** and press Enter or F4-prompt, or another option is to type **DSPDEVD DEVD(device-name)**, then you can skip steps c and d.
- The Display Device Descriptions (DSPDEVD) display appears (Figure 6-15 on page 166).
- Type the device name of the NWSH and press Enter.
- Two more parameters appear, **Option** and **Output**:
 - On the Option parameter, you can type ***ALL**, ***BASIC**, ***STGRSC**, or ***VRTETHRSC** for a NWSH device. If you type ***ALL**, you see all options. On the ***BASIC** option, you can page down to see the other parameters and press Enter to see the other options.
 - On the Output parameter, you can type ***** to display it on the display or ***Print** to create a spoolfile.

► **Change network server host adapter using a CL command.**

To change a device description, follow the steps (Figure 6-15 on page 166):

- On a 5250 command line, type **WRKDEVD *NWSH** and press Enter. Another option is to type **CHGDEVNWSH DEVD(device-name)** and press F4-prompt, then you can skip steps b and c.
- The Change Device Desc (NWSH) (CHGDEVNWSH) display appears.

- c. Scroll down to the NWSH you want to change and type 2 (Change) on the option column (Opt) in front of the NWSH and press Enter.
- d. As with iSeries Navigator, you cannot change all parameters when the NWSH is active. The parameters Online at IPL (ONLINE), Message queue (MSGQ), and Recovery Limits (CMNRCYLMT) can be changed.

Work with Device Descriptions				System:	ITCDEM02
Position to		Starting characters			
Type options, press Enter.					
2=Change		3=Copy	4=Delete	5=Display	6=Print
8=Work with status		7=Rename			
		9=Retrieve source			
Opt	Device	Type	Text		
2	NWSH	*NWSH	Network Server Host Adapter		
					Bottom
Parameters or command					
===>					
F3=Exit	F4=Prompt	F5=Refresh	F6=Create	F9=Retrieve	F12=Cancel
F14=Work with status					

Figure 6-15 WRKDEVD display option 2

► **Start a network server host adapter using CL command.**

To start a network server host adapter (NWSH), follow the steps (Figure 6-16 on page 167):

- a. On a 5250 command line, type WRKCFGSTS *DEV *NWSH and press Enter.
- b. The Work with Configuration Status display appears.

- c. Scroll down to the NWSH you want to start and type 1 (Vary on) on the option column in front of the NWSH and press Enter. This only works if the status is Varied off. The status will flow from Varied off to Vary on pending to Active.

```

Work with Configuration Status                                ITCDEM02
                                                            02/24/06 09:00:50
Position to . . . . . Starting characters
Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status   13=Work with APPN status...

Opt Description      Status      -----Job-----
NH1LIN07            VARIED OFF
NH2LIN07            VARIED OFF
NH3LIN07            VARIED OFF
1  NWSH              VARIED OFF
   TEST              VARIED OFF

Parameters or command
===>
F3=Exit  F4=Prompt  F12=Cancel  F23=More options  F24=More keys
Bottom

```

Figure 6-16 WRKCFGSTS display option 1

► **Stop a network server host adapter using CL command.**

To stop a network server host adapter (NWSH), follow the steps:

- a. On a 5250 command line, type **WRKCFGSTS *DEV *NWSH** and press Enter.
- b. The Work with Configuration Status display appears (Figure 6-17 on page 168).
- c. Scroll down to the NWSH you want to stop and type 2 (Vary off) on the option column in front of the NWSH and press Enter. This only works if the status is Active or Vary on pending. The status will flow from Active to Vary off pending to Varied off.

```

Work with Configuration Status                                ITCDEM02
                                                           02/24/06 09:12:09
Position to . . . . . Starting characters
Type options, press Enter.
  1=Vary on   2=Vary off   5=Work with job   8=Work with description
  9=Display mode status   13=Work with APPN status...

Opt  Description      Status      -----Job-----
    NH1LIN07          VARIED OFF
    NH2LIN07          VARIED OFF
    NH3LIN07          VARIED OFF
  2   NWSH             ACTIVE
    TEST              VARIED OFF

Parameters or command
===>
F3=Exit  F4=Prompt  F12=Cancel  F23=More options  F24=More keys
Bottom

```

Figure 6-17 WRKCFGSTS display option 2

► **Delete a network server host adapter using CL command.**

- a. On a 5250 command line, type `WRKDEVD *NWSH` and press Enter. Another option is to type `DLTDEVD DEVD(device-name)` and press Enter. In this case, you do not get a confirm display, but you can skip steps b and c.
- b. The Work with Device Descriptions display appears (Figure 6-18 on page 169).
- c. Scroll down to the NWSH you want to change and type 4 (Delete) on the option column to the left of the NWSH and press Enter. The NWSH must have the status Varied off, otherwise a message appears saying it is in use.
- d. The Confirm Delete of Device Descriptions display appears (Figure 6-19 on page 169). Press Enter on this display and a message appears stating the object is deleted.

```

Work with Device Descriptions
System: ITCDEM02

Position to . . . . . Starting characters

Type options, press Enter.
  2=Change  3=Copy  4=Delete  5=Display  6=Print  7=Rename
  8=Work with status  9=Retrieve source

Opt Device      Type      Text
  NH1LIN07    *NWSH    NWSH 1 on LIN07
  NH2LIN07    *NWSH    NWSH 2 on LIN07
  NH3LIN07    *NWSH    NWSH 3 on LIN07 using class B addresses
  4  NWSH      *NWSH    Network Server Host Adapter
    TEST      *NWSH

Parameters or command
===>
F3=Exit  F4=Prompt  F5=Refresh  F6=Create  F9=Retrieve  F12=Cancel
F14=Work with status

Bottom

```

Figure 6-18 WRKDEVD display option 4

```

Confirm Delete of Device Descriptions

Press Enter to confirm your choices for 4=Delete.
Press F12 to return to change your choices.

Opt Device      Type      Text
  4  NWSH      *NWSH    Network Server Host Adapter

F12=Cancel

Bottom

```

Figure 6-19 Confirm Delete of Device Description display

6.5 Manage service processor server configuration

As we have described previously, the service processor network server configuration (NWSCFG subtype SRVPRC) is used to configure attributes of the Service Processor (SP) of an xSeries or of the Management Module (MM) of a IBM BladeCenter server. A very important function is the initialize option, which we outline first in detail. After this, we outline the common manageable tasks.

6.5.1 Initialize options for user ID and Password on SP/MM

The User ID and Password on the Service Processor on xSeries or Management Module on a BladeCenter can be changed and put in sync from the iSeries using iSeries Navigator or a CL command. Normally, you use initialize for a new Service Processor after you have renamed the default user ID and password on the Service Processor or Management Module. There are four options on the initialize window of the Service Processor object, but only three of them are supported at the time of writing this IBM Redbook, which are INITIALIZE, SYNCHRONIZE, and CHANGE.

Note: The password has a minimum of five characters.

Here is the explanation of what these three options do:

INITIALIZE does the following (Figure 6-20):

- ▶ Login will be done with the user ID and password from the Service Processor Object. At first creation of the Service Processor object, the user ID and Password are set to “default”, which represents USERID and PASSWORD.
- ▶ After successful verification with the Service Processor on the xSeries or BladeCenter, the first listed in the user profile list on the Service Processor is updated with the new supplied user ID and password.
- ▶ Then the Service Processor object is updated with the user ID and password.
- ▶ If it fails in a successful verification with the user ID and Password from the SRVPRC object, logon is tried with the supplied user ID and Password on the initialize window. If this is successful, the SRVPRC object is updated with this user ID and Password:
 - If the second attempt also fails, a message, “CPFC409 - Not Authorized”, returns.

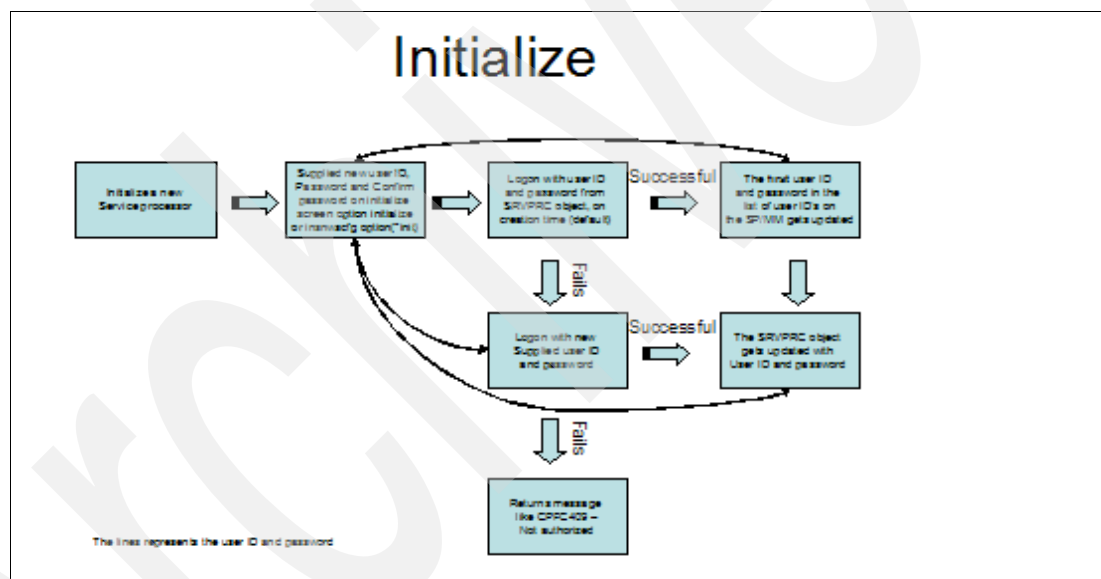


Figure 6-20 Initialize option on initialize window

SYNCHRONIZE does the following (Figure 6-21 on page 171):

- ▶ Login will be done with the provided user ID and password on the initialize window with the synchronize option, so the user ID and password must exist on the service processor of the xSeries or BladeCenter.
- ▶ After successful verification, it changes the SRVPRC object with the user ID and password.
- ▶ If it fails to authenticate, a message, “CPFC409 - Not Authorized”, returns.

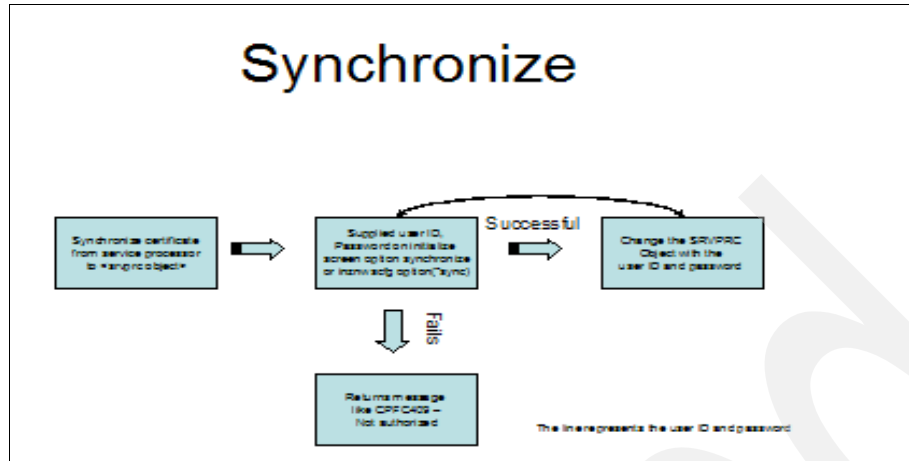


Figure 6-21 Synchronize option on initialize window

CHANGE does the following (Figure 6-22):

- ▶ Login will be done with the user ID and password from the Service Processor Object.
- ▶ After successful verification with the Service Processor on the xSeries or BladeCenter, the first listed in the user profile list on the Service Processor is updated with the new supplied user ID and password.
- ▶ Update the Service Processor Object accordingly.
- ▶ If it fails to authenticate a message, "CPFC409 - Not Authorized", returns.

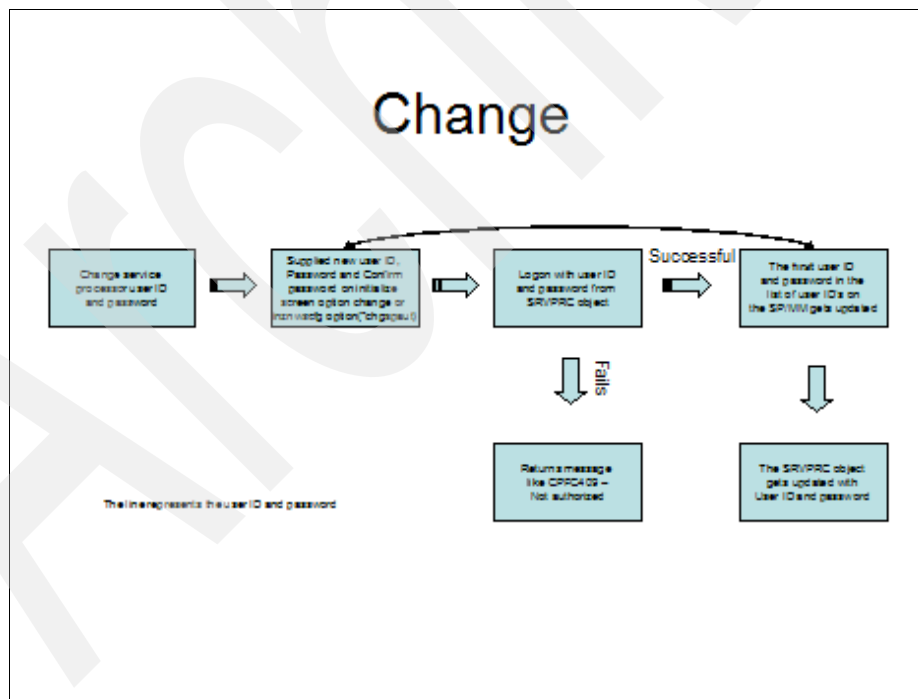


Figure 6-22 Change option on initialize window

6.5.2 Initialize new or changed SP configuration using iSeries Navigator

To initialize a new or changed service processor configuration for a service processor of an xSeries or BladeCenter using iSeries Navigator, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
2. Click **Service Processors**, right-click the service processor at the right pane to create the user ID and Password and select **Initialize**.
3. The Initialize window opens (Figure 6-23), select **Initialize a new processor**.
4. Select **Use specific user and password** and enter an new or existing User ID and Password, and Confirm new password of the service processor.
5. Click **Initialize**.

Note: User ID and Password are both case sensitive, and the *first* user ID and Password in the login profiles list of the Service processor of the xSeries or BladeCenter gets updated.

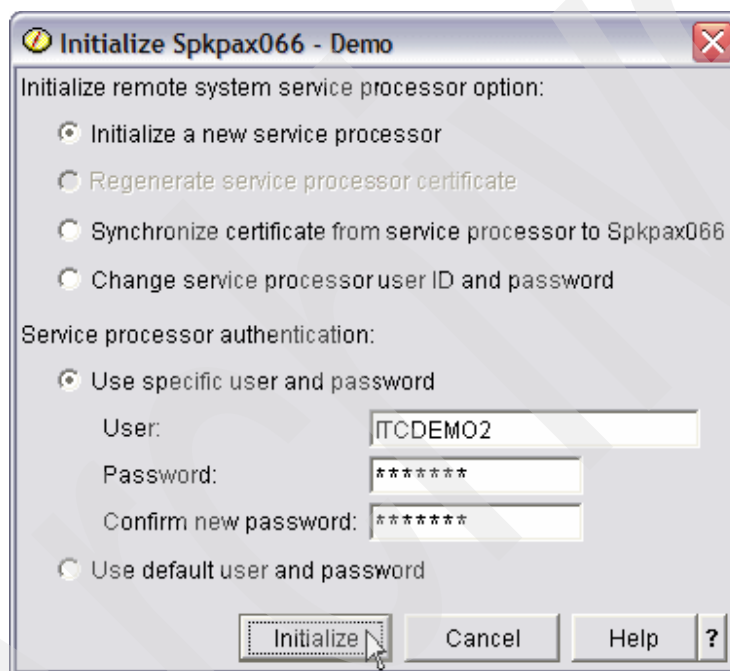


Figure 6-23 Initialize option on Initialize window for existing user

6.5.3 Initialize new user ID and Password on SP/MM using iSeries Navigator

To create a new user ID and password on a service processor of an xSeries or BladeCenter using iSeries Navigator, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
2. Click **Service Processors**, right-click the service processor at the right pane to create the user ID and Password and select **Initialize**.
3. The Initialize window opens (Figure 6-24 on page 173), select **Initialize a new processor**.
4. Select **Use specific user and password** and enter a new User ID, Password, and Confirm new password.
5. Click **Initialize**.

Note: User ID and Password are both case sensitive, and the *first* user ID and Password in the login profiles list of the Service Processor of the xSeries or BladeCenter gets updated.

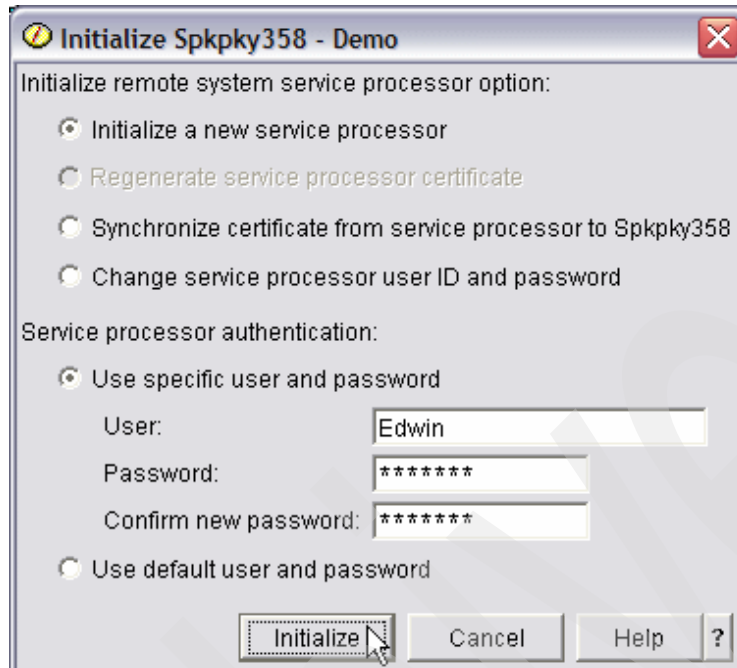


Figure 6-24 Initialize option on Initialize window

6.5.4 Get user ID and password in sync with SP/MM and SRCVPRC object

To get the user ID and password in sync with the Service Processor of the xSeries or BladeCenter with the service processor object using iSeries Navigator, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
2. Click **Service Processors**, right-click the service processor at the right pane to synchronize the user ID and Password.
3. The Initialize window opens (Figure 6-25 on page 174), select **Synchronize certificate from service processor to <Service processor object name>**.
4. Select **Use specific user and password** and specify a User ID and Password, which exist on the service processor of the xSeries or BladeCenter.
5. Click **Initialize**, and the service processor object will be updated accordingly.

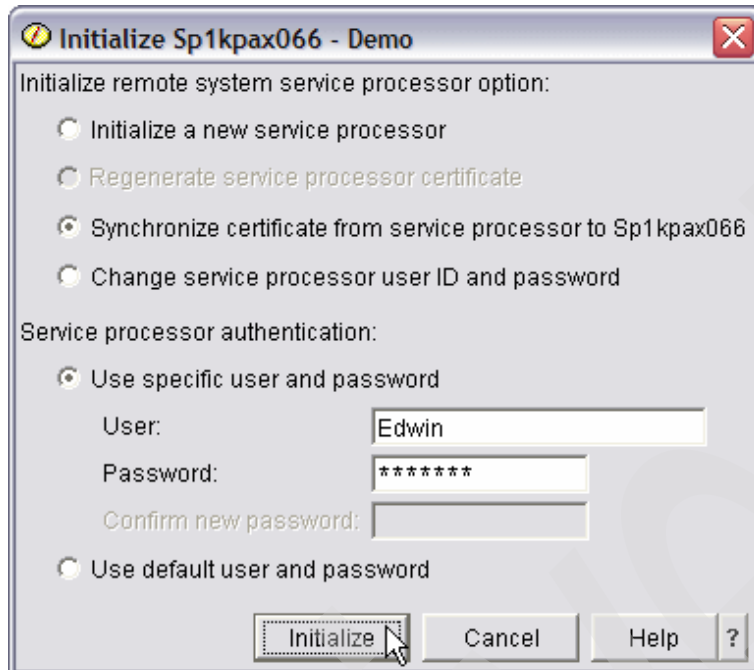


Figure 6-25 Synchronize option on Initialize window

6.5.5 Change service processor password using iSeries Navigator

To change the password on the service processor and SRVPRC object, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
2. Click **Service Processors**, right-click the service processor at the right pane to change the user ID and Password and select **Initialize**.
3. The Initialize window opens (Figure 6-26 on page 175), select **Change service processor user ID and password**.
4. Select **Use specific user and password** and enter an User ID, Password, and Confirm new password.
5. Click **Initialize**.

Note: User ID and Password are both case sensitive, and the *first* user ID and Password in the login profiles list of the Service processor of the xSeries or BladeCenter gets updated.

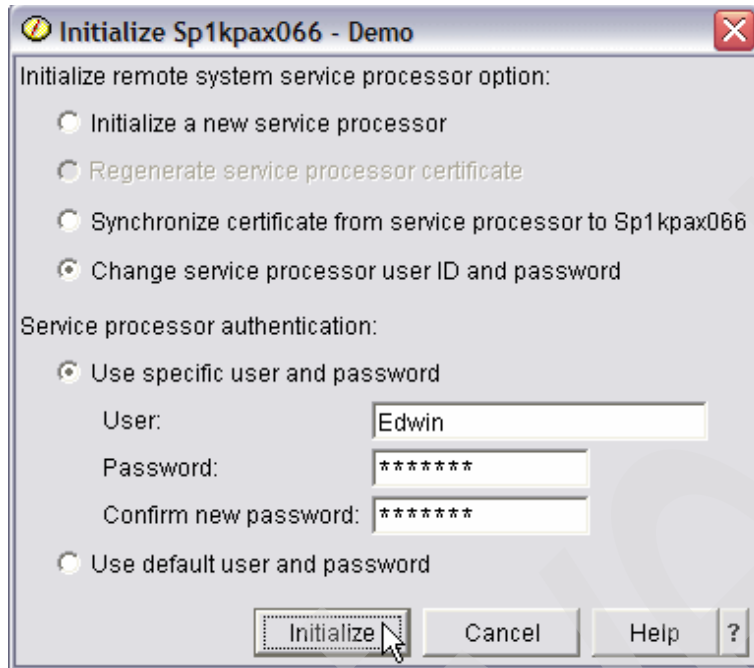


Figure 6-26 Change option on Initialize window

6.5.6 Initialize new or changed SP configuration using CL command

To initialize a new or changed service processor configuration for a service processor of an xSeries or BladeCenter using a CL command, follow the steps:

1. On a 5250 command line, type `INZNWSCFG` and press Enter or F4-prompt. The Initialize NWS Configuration (INZNWSCFG) display appears (Figure 6-27).
2. Type the name of the Service Processor configuration for Network server configuration (NWSCFG). The Change NWS Configuration (CHGNWSCFG) display appears.
3. Type the option `*INIT` for Processing option (Option).
4. Type a new or existing User ID and Password of the Service processor for SP authentication (SPAUT) and press Enter.

```

Initialize NWS Configuration (INZNWSCFG)

Type choices, press Enter.

Network server configuration . . NWSCFG          > SPKPAX066
Processing option . . . . . OPTION          > *INIT
SP authentication:          SPAUT
  User name . . . . .          > ITCDEM02
  User password . . . . .      > passw0rd

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 6-27 INZNWSCFG command with option `*INIT`

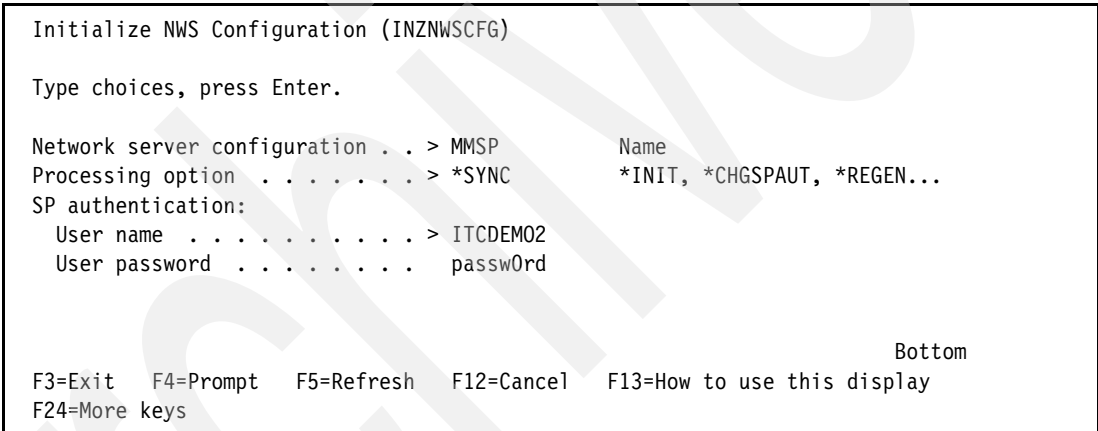
6.5.7 Create new user ID and Password on SP/MM using CL command

To create a new User ID and password on the Service Processor of an xSeries or BladeCenter, follow the same steps described in 6.5.6, “Initialize new or changed SP configuration using CL command” on page 175 with a non-existing User ID and password on the Service Processor.

6.5.8 Synchronize user ID and password with SP/MM and SRCVPRC object

To get the user ID and password in sync with the Service processor of the xSeries or BladeCenter with the service processor object, follow the steps:

1. On a 5250 command line, type `INZNWSCFG` and press Enter or F4-prompt. The Initialize NWS Configuration (INZNWSCFG) display appears (Figure 6-28).
2. Type the name of the Service Processor configuration for Network server configuration (NWSCFG). The Change NWS Configuration (CHGNWSCFG) display appears.
3. Type the option `*SYNC` for Processing option (Option).
4. Type an existing User ID and Password of the Service processor for SP authentication (SPAUT) and press Enter.



```
Initialize NWS Configuration (INZNWSCFG)

Type choices, press Enter.

Network server configuration . . > MMSP           Name
Processing option . . . . . > *SYNC           *INIT, *CHGSPAUT, *REGEN...
SP authentication:
  User name . . . . . > ITCDEM02
  User password . . . . .      passw0rd

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figure 6-28 `INZNWSCFG` command with option `*SYNC`

6.5.9 Change service processor user ID and password using CL command

To change the password on the service processor and SRVPRC object, follow the steps:

1. On a 5250 command line, type `INZNWSCFG` and press Enter or F4-prompt. The Initialize NWS Configuration (INZNWSCFG) display appears (Figure 6-29 on page 177).
2. Type the name of the Service Processor configuration for Network server configuration (NWSCFG). The Change NWS Configuration (CHGNWSCFG) display appears.
3. Type the option `*CHGSPAUT` for Processing option (Option).
4. Type an existing User ID and new Password of the Service processor for SP authentication (SPAUT) and press Enter.

```

Initialize NWS Configuration (INZNWSCFG)

Type choices, press Enter.

Network server configuration . . > MMSP           Name
Processing option . . . . . > *CHGSPAUT       *INIT, *CHGSPAUT, *REGEN...
SP authentication:
  User name . . . . . > ITCDEM02
  User password . . . . .      passw0rd1

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 6-29 INZNWSCFG command with option CHGSPAUT

6.5.10 Manage service processor server configurations using iSeries Navigator

► Create a service processor configuration using iSeries Navigator.

To create a service processor configuration object (NWSCFG subtype SRVPRC), follow these steps:

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Right-click **Service Processors** and select **New Service Processor Configuration**.
- c. The New Service Processor Configuration window opens, on the General tab, enter the following (Figure 6-30 on page 178):
 - i. **SRVPRC Name**.
 - ii. **SRVPRC Description**.
 - iii. There are several ways to discover the enclosure identity using host name or Internet address or just the serial number, but we recommend you use the method of selecting **Use service processor connection to determine remote system enclosure identity**, then select **Internet address**, and enter the Internet address of the service processor of the xSeries or BladeCenter chassis, because when the Internet address is specified, unicast is done, which means that the packets are sent directly to the service processor. Otherwise, when not specifying the host name or Internet address, a broadcast is done, which we do not recommend.
- d. Enter the Serial number at Enclosure identity. The serial number is of the xSeries or BladeCenter Chassis. This can be found on the system itself (small sticker) or using the WEB browser to the Service processor of the xSeries or Blade Center.
- e. Select the **Object authority** and click **OK**.

Important: At creation time, the user ID for the service processor authentication is default. Default is USERID for the user ID and PASSWORD (0 = zero) for the password.

Note: There is only one option on the Security tab, which cannot be changed. So you do not have to select it, because this feature is not available yet at the time of writing this book.

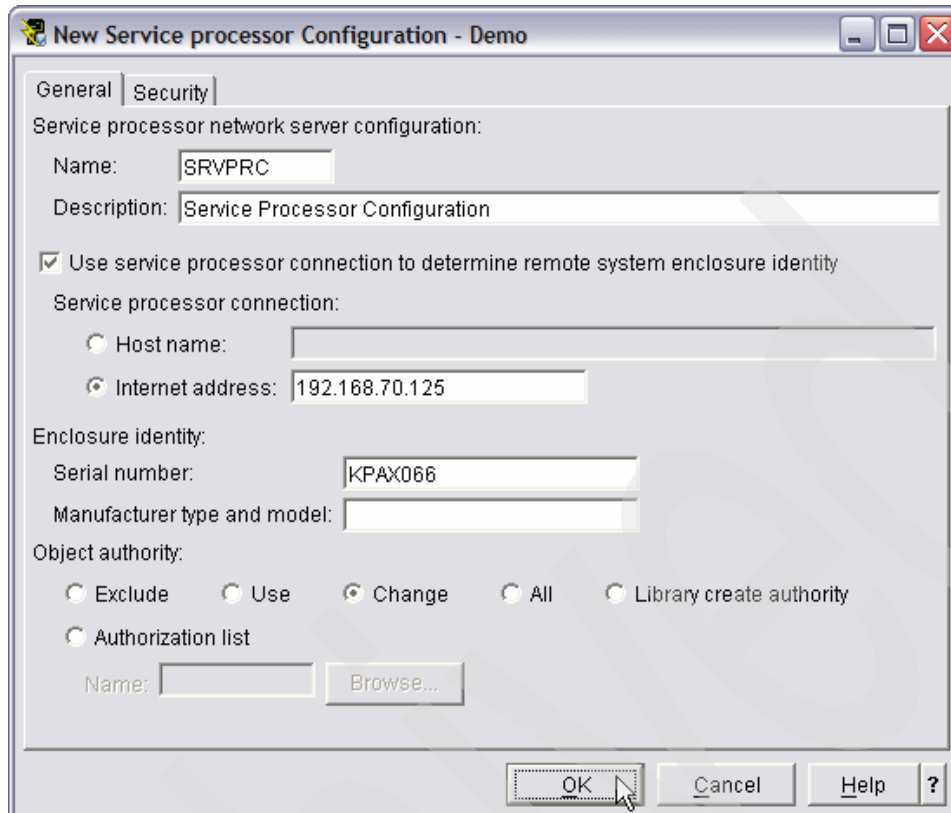


Figure 6-30 General tab New Service Configuration window

► **Create a service processor configuration based on another one using iSeries Navigator.**

To create a SRVPRC based on another one using iSeries Navigator, follow the steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Service Processors** and right-click the Service processor configuration at the right pane, which you want to copy from and select **New Based On**. See Figure 6-31 on page 179.
- Enter a new SRVPRC name and change any other attribute as described in Create a service processor configuration using iSeries Navigator, and click **OK**.

Important: The user ID and password for the service processor authentication will be the same as the service processor configuration user ID and password you copied from.

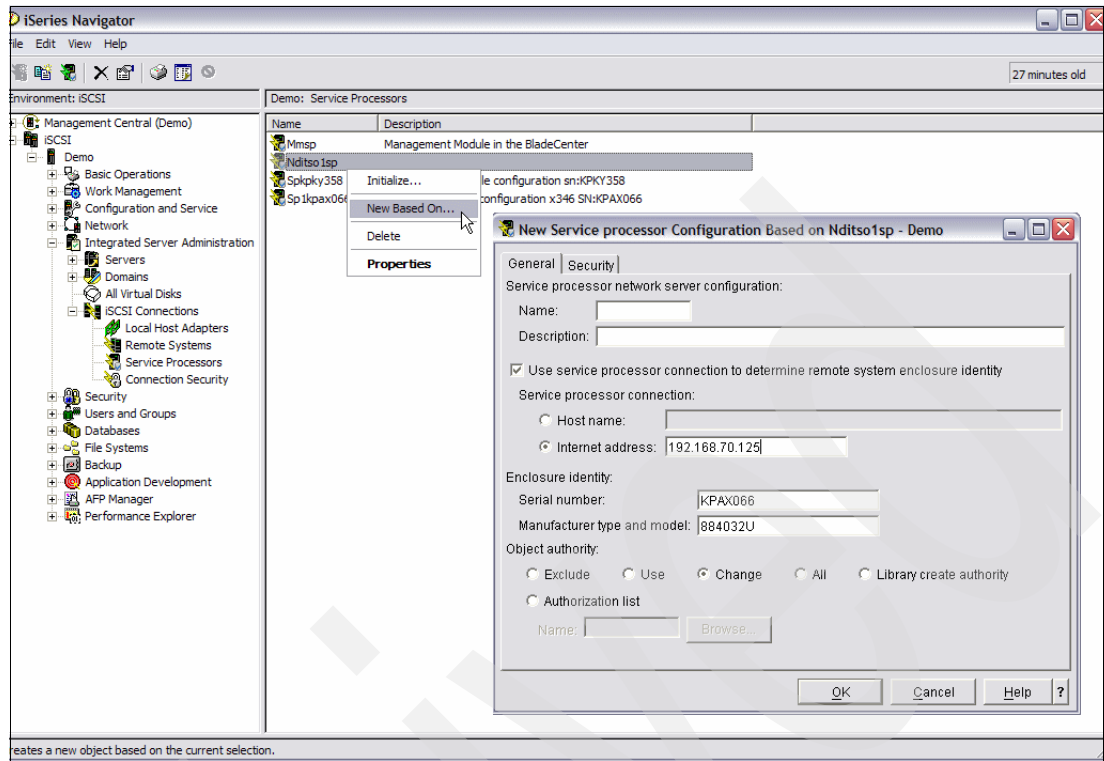


Figure 6-31 New Based on option on a Service processor configuration

► **Display/Change service processor configuration using iSeries Navigator.**

To display or change a SRVPRC configuration, you follow the same steps. The following attributes can be changed:

On the General tab:

- Description
- Service processor connection's host name or Internet address
- Enclosure identity's Serial number, Manufacture type, and model

On the Security tab, you cannot change anything, but you are able to see the User ID and password for the Service processor authentication. To change the user ID and password of the Service Processor, you have to use the Initialize option described in 6.5.1, "Initialize options for user ID and Password on SP/MM" on page 169.

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Click **Service Processors** and right-click **Service processor configuration** at the right pane, which you want to display or change from and select **Properties** (See Figure 6-32 on page 180).

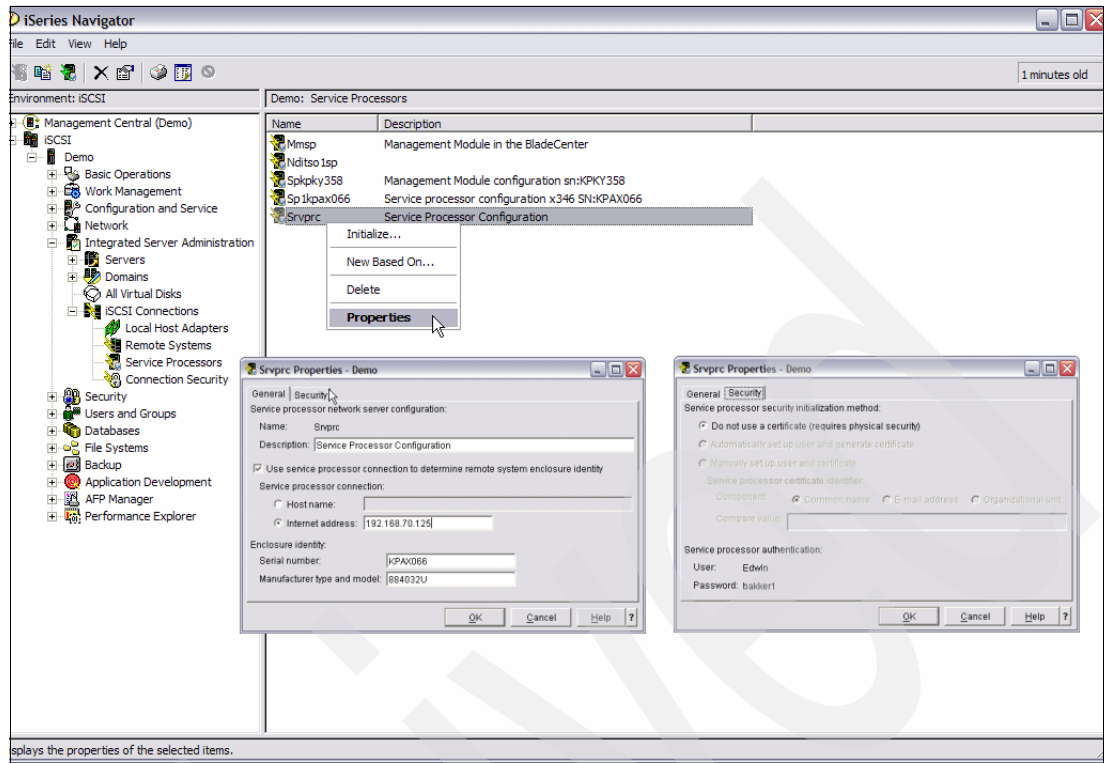


Figure 6-32 Properties Service Processor Configuration

- **Delete a service processor configuration using iSeries Navigator.**
 - a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
 - b. Click **Service Processors**, right-click the **Service processor configuration** at the right pane which you want to delete, and select **Delete**.
 - c. On the Confirm Delete Service Processor configuration window, select **Delete** (Figure 6-33 on page 181).

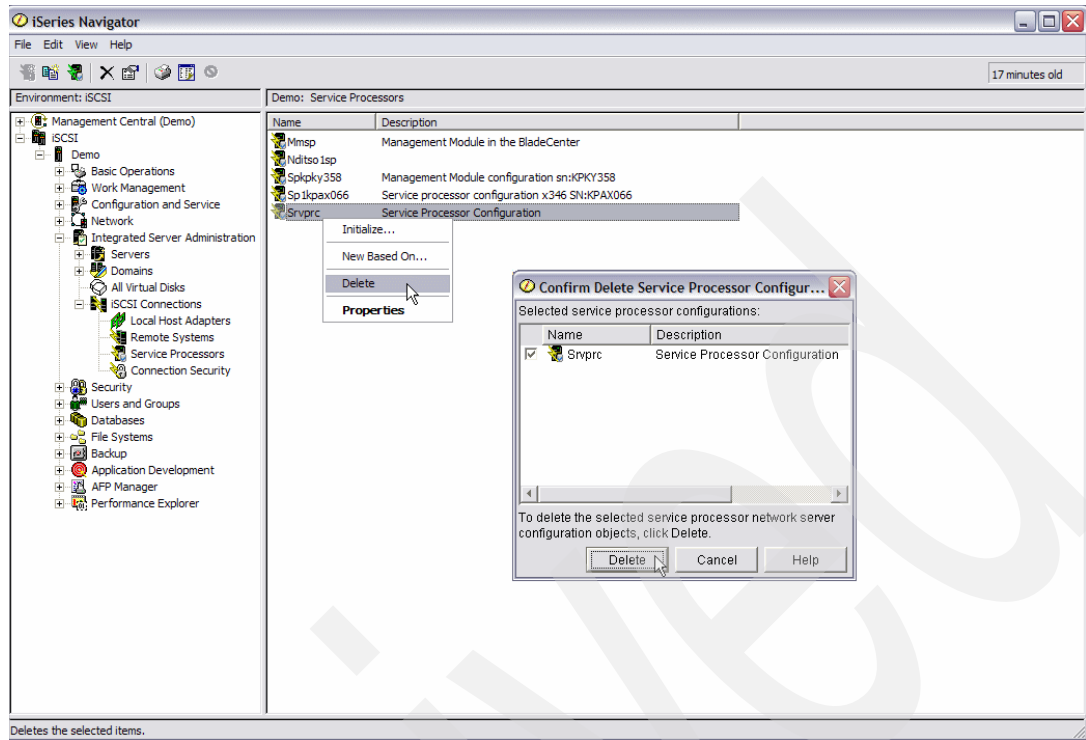


Figure 6-33 Delete Service processor configuration

6.5.11 IP address change for Service Processor

If you plan to change the IP address of the service processor of an xSeries or the Management Module of a BladeCenter chassis in an iSCSI integrated server environment, you must change the Service Processor configuration as well, and you should do an initialize:

► Follow these steps if this is for an SP of an xSeries:

1. Change the IP address in the Service processor of an xSeries using the Web interface:
 - a. Follow steps 1 through 4 described in 6.8.7, "Starting Remote Control on RSA II to see Windows booting" on page 230, then return here to follow the rest of these steps.
 - b. The System Status window appears of the SP showing the status of the xSeries. Click **Network interfaces** under ASM Control, the network interface window appears (Figure 6-34 on page 182).
2. Change the changed IP address of the SP in the Service Processor Configuration used for this SP described in 6.5.10, "Manage service processor server configurations using iSeries Navigator" on page 177 or 6.5.12, "Manage service processor server configurations using CL command" on page 182.
3. Use the initialize option to update the database of IBM Director server described in 6.5.2, "Initialize new or changed SP configuration using iSeries Navigator" on page 172.

► Follow these steps if it concerns an MM of a BladeCenter chassis:

1. Change the IP address in the Management Module of an BladeCenter chassis using the Web interface.
 - a. Follow steps 1 through 4 as described in 6.8.6, "Starting Remote Control on MM to see Windows booting" on page 229, then return here to follow the rest of the steps.

- b. The System Status window of the SP appears showing the status of the xSeries. Click **Network interfaces** under MM Control, the Management Module Network Interfaces window appears (Figure 6-34). Click **External Network Interface (eth0)**.
- c. Change the **Static IP Configuration** according to your needs.

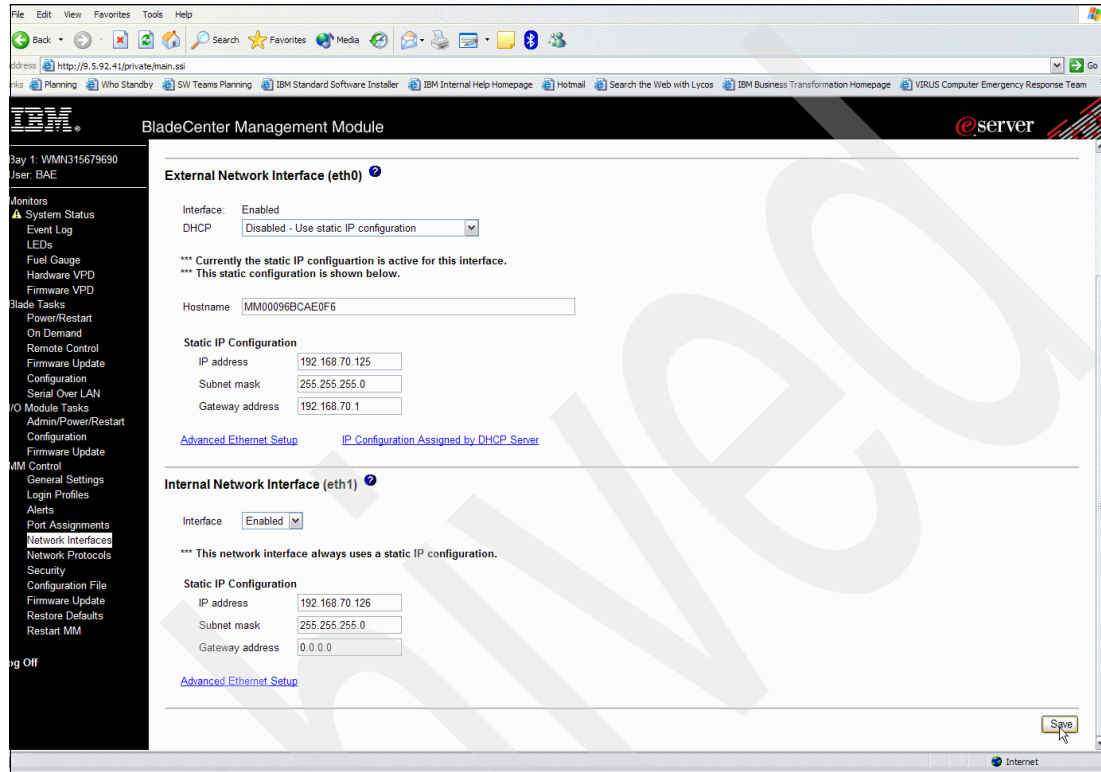


Figure 6-34 Network interface window MM

2. Change the changed IP address of the MM in the Service Processor Configuration used for this MM described in 6.5.10, "Manage service processor server configurations using iSeries Navigator" on page 177 or 6.5.12, "Manage service processor server configurations using CL command" on page 182.
3. Use the initialize option to update the database of IBM Director server described in 6.5.2, "Initialize new or changed SP configuration using iSeries Navigator" on page 172 or 6.5.6, "Initialize new or changed SP configuration using CL command" on page 175.

6.5.12 Manage service processor server configurations using CL command

► Create a service processor configuration using CL command.

To create a service processor configuration (NWSCFG subtype SRVPRC), follow these steps:

- a. On a 5250 command line, type `CRTNWSCFG` and press Enter or F4-prompt.
- b. The Create NWS Configuration (CRTNWSCFG) display appears (Figure 6-35 on page 183).
- c. Type a **Network server configuration** (NWSCFG) name and **Configuration type** (TYPE). In this case, it is *SRVCFG for Service processor configuration and press Enter for more parameters.

- d. Type *NONE for **Initialize** service processor (INZSP), the other two options *MANUAL and *AUTO are not supported at this time. Support for the values might be provided at a future date.
- e. Type *YES for **Enable Unicast** (ENBUNICAST), then the packets are sent directly to the service processor using the value for the parameter Service processor name, otherwise, a broadcast is done. That is why *YES is recommended.
- f. Type *SPINTNETA for **Service processor name** (SPNAME), which is recommended. An Internet address should then be specified for **SP Internet address** (SPINTNETA). A host name can also be specified, then the parameter SP Internet address (SPINITNETA) is not needed.
- g. Type the serial number of the xSeries or BladeCenter chassis for **Serial number** (EID). Manufacturer type and model parameter are optional, so not required.
- h. Type a Description for **Text 'description'** (TEXT) and press Enter. You will see a message at the bottom of the display confirming that the Network server configuration is created.
- i. To change the **Authority** (AUT), you have to press F10-Additional parameters.

```

                                Create NWS Configuration (CRTNWSCFG)

Type choices, press Enter.

Network server configuration . . NWSCFG          > SRVPRC
Configuration type . . . . . TYPE              > *SRVPRC
Initialize service processor . . INZSP          > *NONE
Enable unicast . . . . . ENBUNICAST            > *YES
Service processor name . . . . . SPNAME        > *SPINTNETA

SP internet address . . . . . SPINTNETA        > '192.168.70.125'
Enclosure identifier:      EID
  Serial number . . . . .                    > KPAX066
  Manufacturer type and model .
Text 'description' . . . . . TEXT              > 'Service processor configuratio
n'

                                Additional Parameters

Authority . . . . . *CHANGE          Name, *CHANGE, *ALL, *USE...

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display   F24=More keys

```

Figure 6-35 CRTNWSCFG display for Service Processor configuration

► **Display a service processor configuration using CL command.**

To display a Service processor configuration follow the steps:

- a. On a 5250 command line, type DSPNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG(*Service processor name*) and put option 5 (Display) on the option column in front of the SRVPRC to display.
- b. The Display NWS Configuration (DSPNWSCFG) display (Figure 6-36 on page 184) appears.
- c. Type the name of the SRVPRC and press Enter.

- d. Two more parameters appear, **Option** and **Output**.
 - iv. On the Option parameter you can type *ALL, *BASIC, or *SRVPRC for a SRVPRC configuration. If you type *ALL, you will see all options. On the *SRVPRC option, you can page down to see the other parameters.
 - i. On the Output parameter, you can type * to display it on the display or *Print to create a spoolfile.

► **Change service processor configuration using CL command.**

To change a Service processor configuration, follow the steps:

- a. On a 5250 command line, type CHGNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG(*Service processor name*) and put option 2 (Change) on the option column in front of the SRVPRC to change.
- b. The Change NWS Configuration (CHGNWSCFG) display appears.
- c. Type the name of the SRVPRC and press Enter.
- d. After each enter the parameters appears for you to change. When bottom appears at the right down corner the last parameter is shown the next enter will make the changes. You cannot change the user ID and password for the service processor, you have to use the command INZNWSCFG described in 6.5.9, "Change service processor user ID and password using CL command" on page 176.

Change NWS Configuration (CHGNWSCFG)

Type choices, press Enter.

Network server configuration . . . NWSCFG	>	SRVPRC
Initialize service processor . . . INZSP	>	*NONE
Enable unicast ENBUNICAST	>	*YES
Service processor name SPNAME	>	*SPINTNETA
SP internet address SPINTNETA	>	'192.168.70.125'
Enclosure identifier: EID		
Serial number		KPAX066
Manufacturer type and model		884032U
Text 'description' TEXT		'Service processor configuratio
n'		
Bottom		
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display		
F24=More keys		

Figure 6-36 CHGNWSCFG display for SRVPRC

► **Delete a service processor configuration using CL command**

To delete a Service processor configuration, follow the steps:

- a. On a 5250 command line, type DLTNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG(*Service processor name*) and put option 4-Delete on the option column in front of the SRVPRC to delete.
- b. The Delete NWS Configuration (DLTNWSCFG) display appears (Figure 6-37 on page 185).
- c. Type the name of the SRVPRC and press Enter.
- d. A message appears that the object is deleted.

Delete NWS Configuration (DLTNWSCFG)

Type choices, press Enter.

Network server configuration . . > SRVPRC	Name
	Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 6-37 DLTNWSCFG display for SRVPRC

6.6 Manage remote system server configurations

As we have previously described, the remote system network server configuration (NWSCFG subtype RMTSYS) object is used to configure attributes of an iSCSI attached xSeries or Blade within an IBM BladeCenter chassis.

6.6.1 Manage RMTSYS object using iSeries Navigator

The following steps show how to manage RMTSYS objects using iSeries Navigator:

► **Create a remote system configuration object using iSeries Navigator.**

To create a remote system configuration object (NWSCFG subtype RMTSYS), follow these steps:

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Right-click **Remote Systems** and select **New Remote System Configuration**.
- c. The New Remote System Configuration window opens (Figure 6-38 on page 186), on the General tab, enter the following:
 - i. **RMTSYS Name**.
 - ii. **RMTSYS Description**.
 - iii. Select the **Service processor configuration** using the pull-down list, you can select **Properties** to see the properties of the Service processor configuration.
 - iv. Select **Use enclosure identity from Service processor configuration** when the remote system is an xSeries. If this concerns a Blade in a BladeCenter, you should select **Use the following values** and specify the Serial Number of the Blade for this configuration. The serial number in the Service processor configuration to which it references is the serial number of the BladeCenter chassis and not the Blade itself!
 - v. Select **Object authority** and select the Boot Parameters tab.

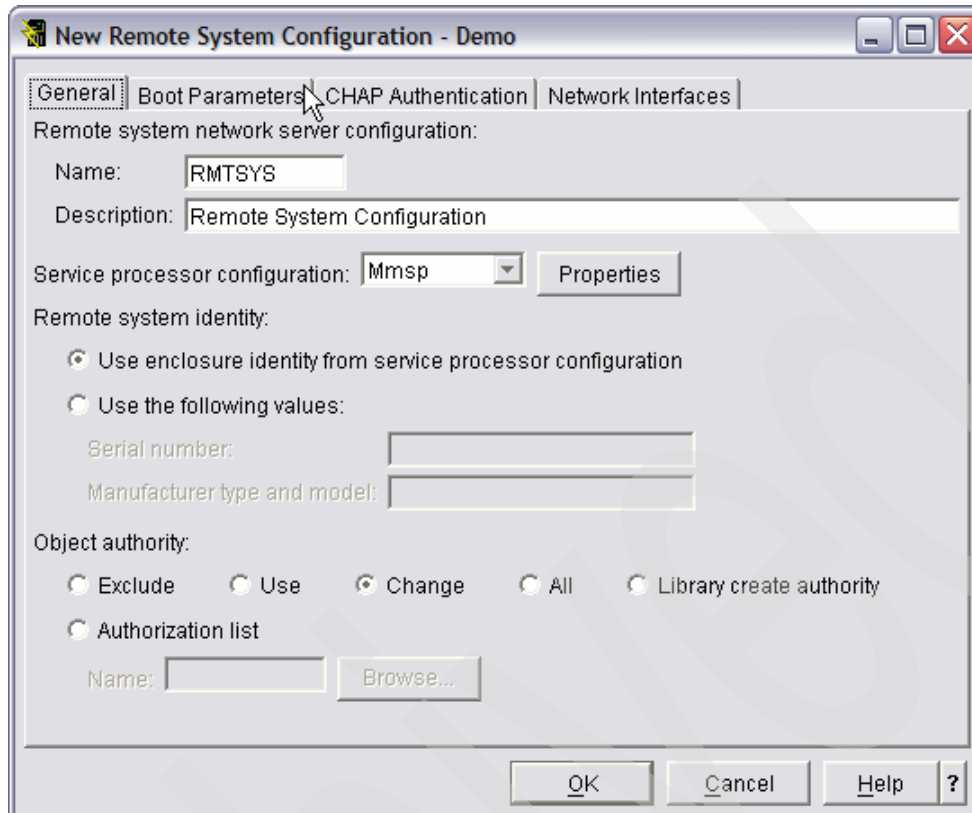


Figure 6-38 General tab New Remote System Configuration window

- d. On the Boot Parameters tab, enter the following:
 - i. Select **Dynamically Delivered to remote system via DHCP**, leave the **Vendor ID** and **Alternate Client ID** to the default values, which are Default and Adapter. There is also an option to manually configure it on the remote system, but we definitely do not recommend it. The Vendor ID will be set to IBM ISAN on the initiator adapter in the xSeries or Blades, which is the only value to use. The Alternate Client ID cannot be used at the time of writing this book.
 - ii. If you have more than one iSCSI initiator on the xSeries or use the two ports on an daughter card on a Blade, you must specify which initiator will be doing the boot. So you must specify the **Bus**, **Device**, and **Function** doing the boot. These values can be found with the CTRL-Q utility during the boot of the xSeries or Blade (Figure 6-39). Select the CHAP Authentication tab.

Select Host Adapter								
Adapter	Boot Mode	I/O Address	Slot	Bus	Device	Function	MAC Address	
QMC4052	DHCP	5100	01	06	01	1	00-C0-DD-07-59-8A	
QMC4052	Disable	5300	01	06	01	3	00-C0-DD-07-59-BC	

Figure 6-39 CTRL-Q utility Bus, Device, and Function

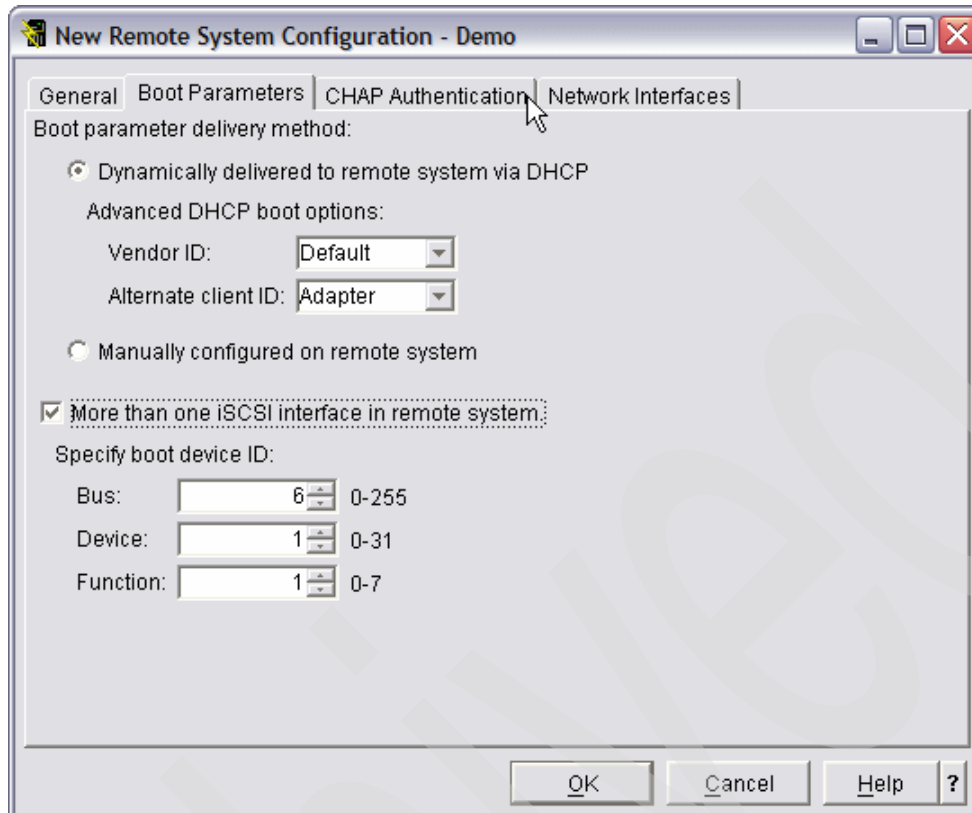


Figure 6-40 Boot Parameters tab New Remote System Configuration window

- e. On the CHAP Authentication tab (Figure 6-41 on page 188), enter and select the following:
 - i. Select **Use the following values for CHAP authentication** if you want to use CHAP (we recommend this).
 - ii. Enter a CHAP Name or leave the default to use the remote system configuration name instead.
 - iii. You could either select **Generate CHAP secret once** (recommended) or specify one yourself by selecting **Specific** CHAP secret. You can use special characters as well and select the Network Interfaces tab.

Important: This must match the entry in the iSCSI adapter. The CTRL-Q utility can be used to set this. The CHAP user ID and password are case sensitive. Even if the CHAP name is created with the First letter uppercase and the rest lowercase, it is actually all uppercase. You can check this within a 5250 screen, displaying the RMTSYS configuration.

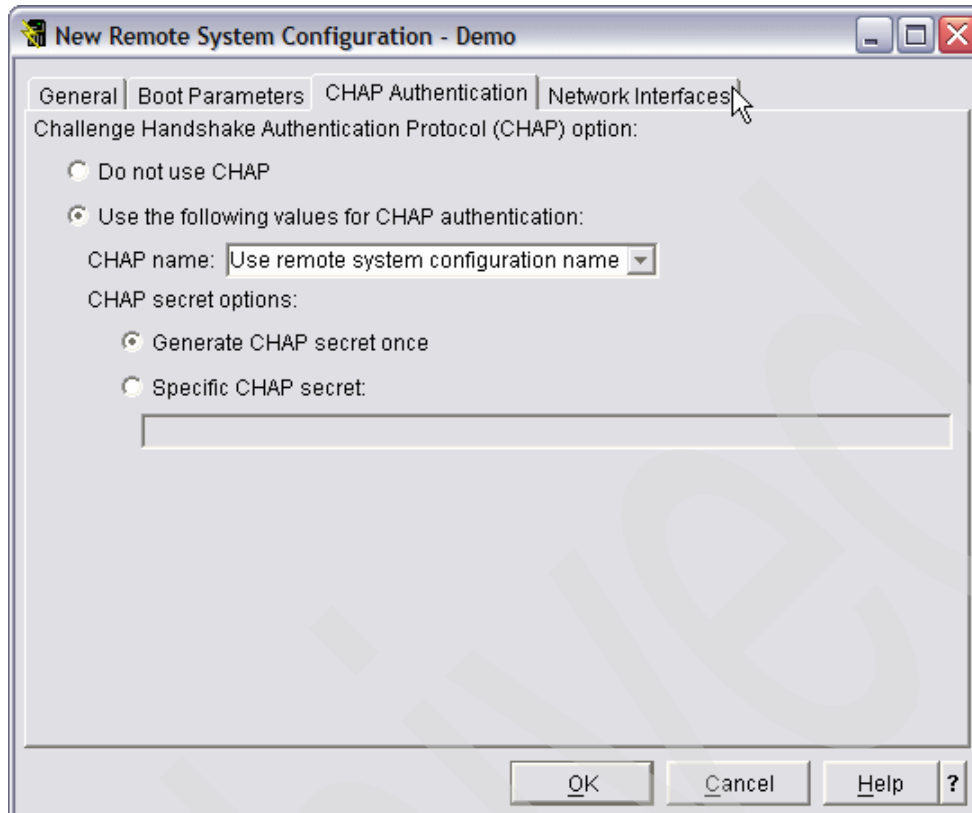
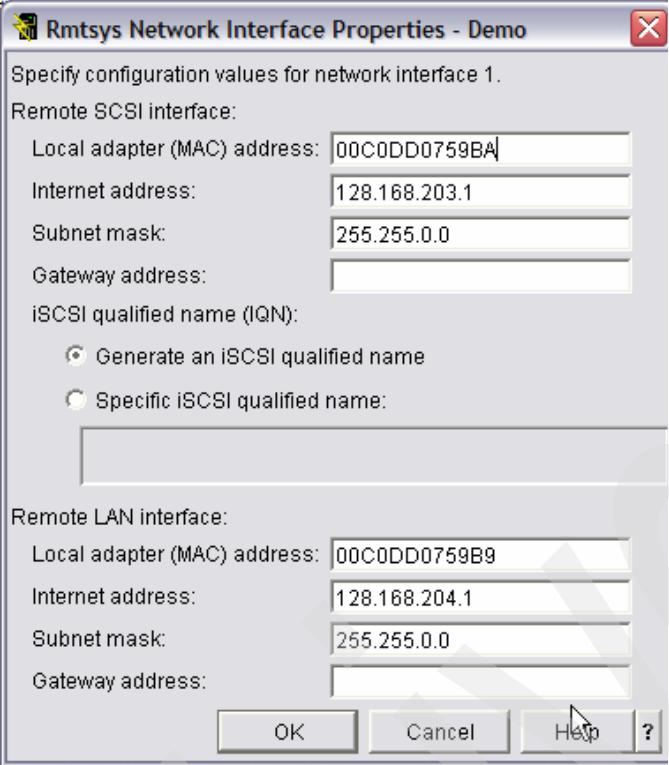


Figure 6-41 CHAP Authentication tab New Remote System Configuration

- f. On the Network Interfaces tab, enter and select the following:
 - i. Select **ADD** to enter the values for the remote network interface.
 - ii. The Rmtsys Network Interface Properties window appears (Figure 6-42 on page 189).
 - iii. Enter the **MAC-Address** of the iSCSI part of the card in the xSeries or Blade.
 - iv. Enter the Internet address and **Subnet Mask** for the iSCSI part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH this integrated server will use.
 - v. The **Gateway Address** can be left blank, it is not used at the time of writing this book.
 - vi. Select **Generate an iSCSI qualified name** (recommended), you can specify your own. This is described in RFC3722.
 - vii. Enter the **MAC-Address** of the LAN part of the card in the xSeries or Blade.
 - viii. Enter the Internet address and Subnet Mask for the LAN part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH, which is used by the integrated server.
 - ix. The **Gateway Address** can be left blank, it is not used at the time of writing this book, and select **OK** on the Rmtsys Network Interface Properties window.
 - x. Select **OK** again on the New Remote System Configuration window.

Note: The iSCSI card has two MAC-Addresses, one for iSCSI and one for LAN connection, which is also called TOE.



Rmtsys Network Interface Properties - Demo

Specify configuration values for network interface 1.

Remote SCSI interface:

Local adapter (MAC) address: 00C0DD0759BA

Internet address: 128.168.203.1

Subnet mask: 255.255.0.0

Gateway address:

iSCSI qualified name (IQN):

☒ Generate an iSCSI qualified name

☐ Specific iSCSI qualified name:

Remote LAN interface:

Local adapter (MAC) address: 00C0DD0759B9

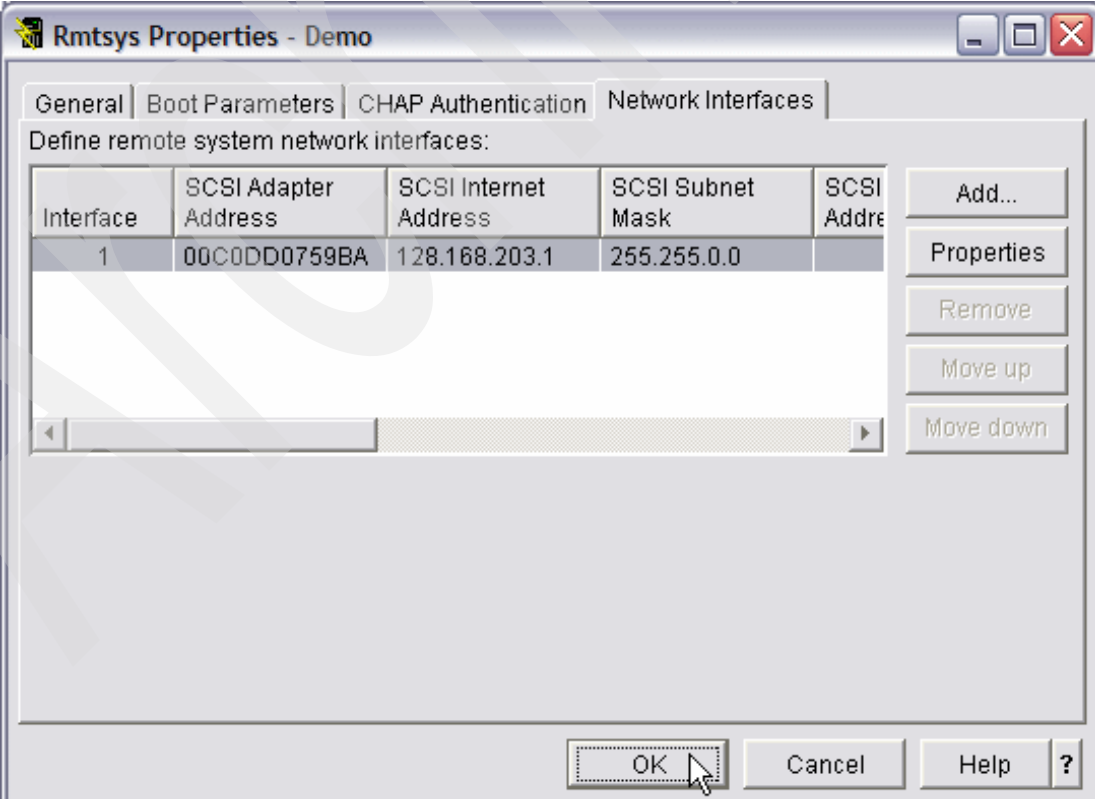
Internet address: 128.168.204.1

Subnet mask: 255.255.0.0

Gateway address:

OK Cancel Help ?

Figure 6-42 Rmtsys Network Interface Properties window



Rmtsys Properties - Demo

General | Boot Parameters | CHAP Authentication | **Network Interfaces**

Define remote system network interfaces:

Interface	SCSI Adapter Address	SCSI Internet Address	SCSI Subnet Mask	SCSI Address
1	00C0DD0759BA	128.168.203.1	255.255.0.0	

Add... Properties Remove Move up Move down

OK Cancel Help ?

Figure 6-43 Network Interface tab New Remote System Configuration

To create a RMTSYS based on another one using iSeries Navigator, follow the steps (Figure 6-44):

- g. Enter a new RMTSYS **name** and change any other attribute as described in Create a remote system configuration using iSeries Navigator and select **OK**.
- h. Remember that every attribute is copied, including the CHAP authentication and the network interfaces.

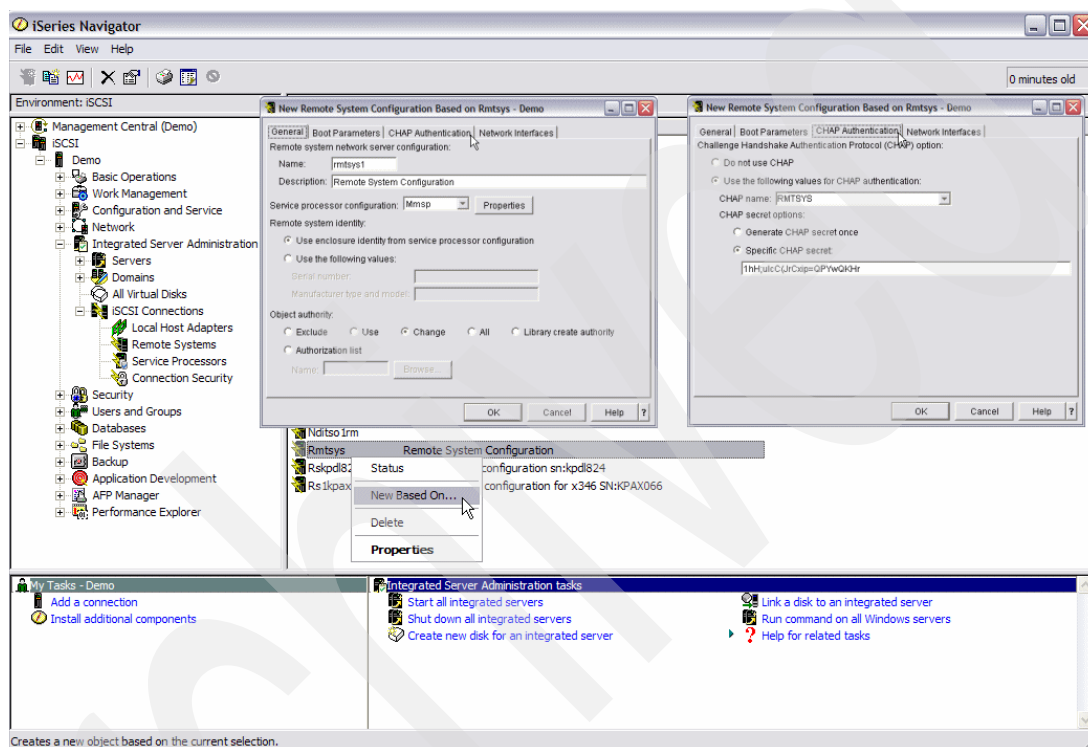


Figure 6-44 New Based on option for a Remote System Configuration

- **Display/Change remote system configuration properties using iSeries Navigator.**

To display or change a RMTSYS configuration, you follow the same steps (Figure 6-45 on page 191). You can change all attributes besides the name of the RMTSYS configuration. You can even display/change the attributes of the service processor configuration by clicking **properties** on the general tab next to the pull-down menu of Service processor configuration. In order to display/change one of the Network interfaces, you must select it and click **properties** on the Network interfaces tab:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Service Processors** and right-click **Service processor configuration** at the right pane, which you want to display or change from, and select **Properties**.

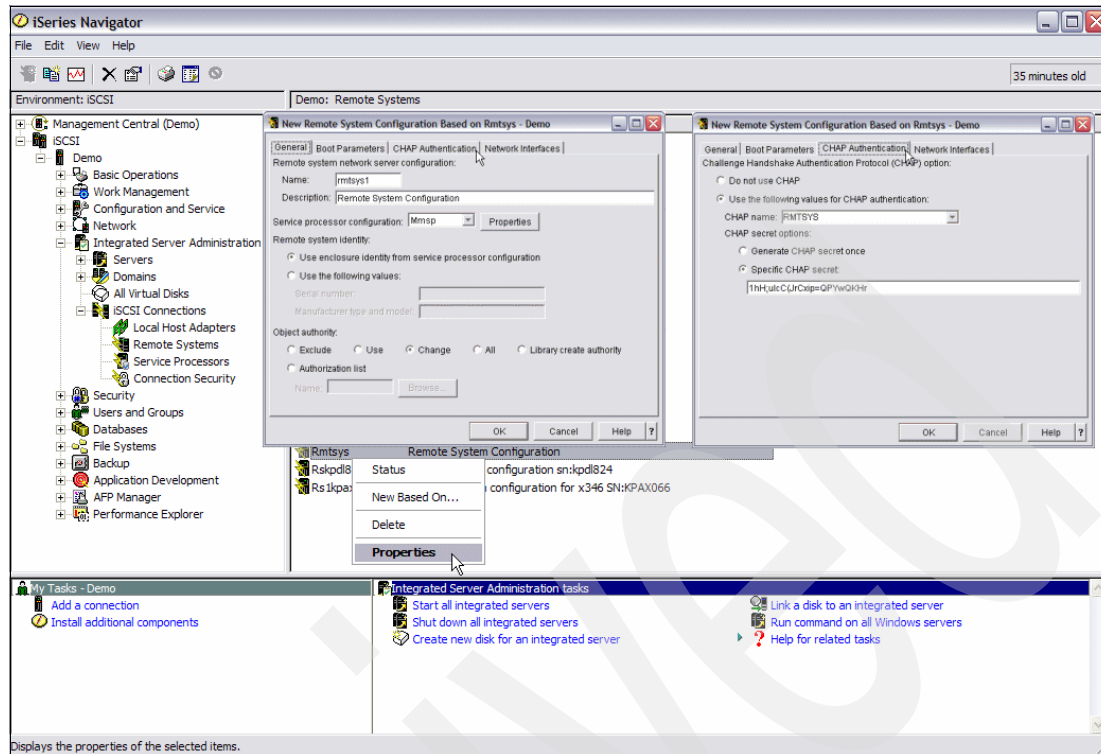


Figure 6-45 Properties Remote System Configuration

► **Display remote system status (xSeries or BladeCenter) using iSeries Navigator.**

The hardware status of the xSeries or BladeCenter server can be retrieved. This comes in handy to check if IBM Director server is able to reach and retrieve the status of the server.

On the status window, there is a button “error details” and in case of a “Status retrieval error”, it displays a message why. Other hardware status could be:

- Powered off (or state unknown)
- System power on before POST
- Booting (in POST)
- Stopped booting with POST error
- Booting flash or system partition
- Booting operating system
- In operating system
- CPU held in reset
- Powered on
- Unknown (timeout)
- Status retrieval error

You can also monitor the status during the vary on of the iSCSI integrated server by pressing **refresh** on the status window. So you can follow which state the server is in.

You can also set the **Status Timeout**, the time allowed to retrieve the information. The default is 30 seconds, and the maximum is 300 seconds.

Among the status of the server, there is also the following information:

Last collected	Date/time the last status was retrieved
Service processor Internet address	Internet address of xSeries or Blade center
Serial number	Serial number of the xSeries or BladeCenter chassis
Manufacture type and model	Manufacture type and model of the xSeries or BladeCenter chassis

To display the status of a remote server, follow the steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Remote Systems** and right-click the **Remote system configuration** at the right pane to check the status, select **Status**. See Figure 6-46.

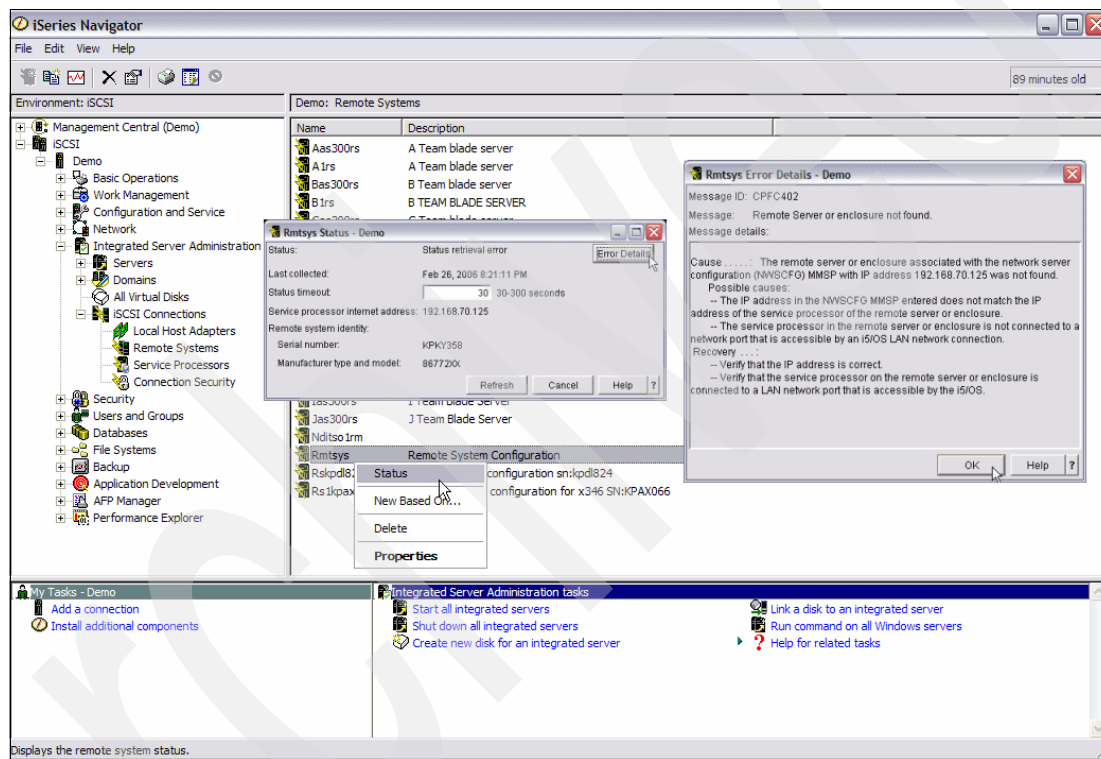


Figure 6-46 Status Remote system and error details

► **Delete a remote system configuration using iSeries Navigator:**

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Remote systems** and right-click the **Remote system configuration** at the right pane which you want to delete, and select **Delete**.
- On the Confirm Delete Remote System configuration window (Figure 6-47 on page 193), select **Delete**.

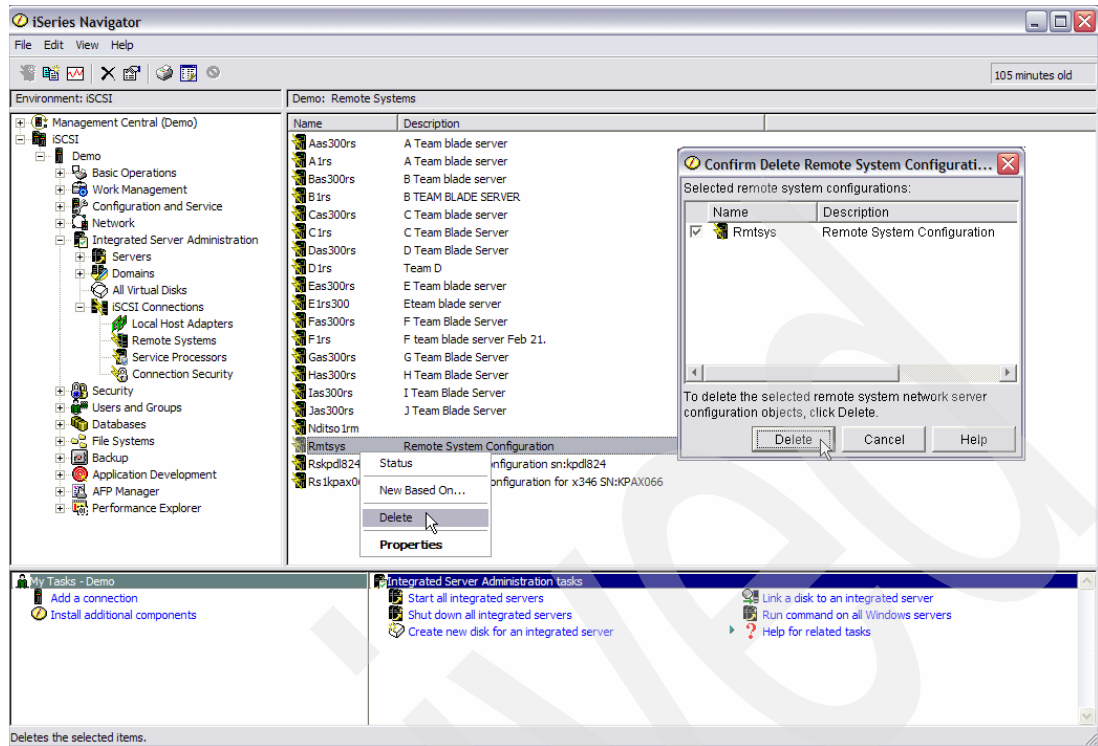


Figure 6-47 Delete Remote System configuration

► Change the CHAP authentication using iSeries Navigator

The CHAP authentication can be deferred from the remote system configuration on the CHAP authentication page displaying the properties. If you want to change the CHAP authentication, this has to be done on two places: 1. Within the Remote System configuration and 2. Within the CTRL-Q utility on the xSeries server or Blade within the BladeCenter chassis to configure the iSCSI card.

To change the CHAP authentication, follow the steps:

- Change the remote system configuration's CHAP authentication page as described in Display/Change remote system configuration properties using iSeries Navigator with the new values for the CHAP name and CHAP secret.
- Start the CTRL-Q utility on the xSeries server or the correct Blade in a BladeCenter. You can do this on the console of the xSeries or BladeCenter or within the remote console utility using a WEB browser pointing to the service processor of the xSeries or BladeCenter described in 6.8.6, "Starting Remote Control on MM to see Windows booting" on page 229 for an xSeries and in 6.8.7, "Starting Remote Control on RSA II to see Windows booting" on page 230 for a BladeCenter.

Note: For BladeCenter, select first the right Blade for the Keyboard, Video, and Mouse (KVM) to use the monitor, keyboard, and mouse by selecting the monitor icon at the left top of the Blade.

- Start the xSeries or Blade with the power on button or within the WEB browser pointing to the service processor. If it is already up and running, you can also restart the server. So Power-On Self Test (POST) will be started on the server.

- ii. Some time after the eServer logo has been displayed, the QLogic BIOS prompt shows. It reads "Press <CTRL-Q> for Fast!UTIL". At this time, issue Ctrl-Q. See Figure 6-48 on page 195 and Figure 6-49 on page 195. If you are using the remote console utility, select **Ctrl** at the toolbar with the MM of the BladeCenter. For the xSeries, you have to configure this first using preferences.
- iii. When this is successful, it reads "CTRLQ> Detected, Initialization in progress, Please wait", and the QLogic Fast!UTIL menu appears.
- iv. If more than one adapter is installed, select the one to change the CHAP authentication using the up and down arrow keys and press Enter.
- v. The selected adapter is shown at the top and another pane, Fast!UTIL options. Select **Configuration Options** and press Enter.
- vi. The Configuration Options window appears, select **iSCSI Boot settings** using the up and down arrow keys and press Enter.
- vii. The iSCSI Boot settings window appears, select **Primary Boot Device Settings** using the up and down arrow keys and press Enter.
- viii. The Primary Boot Device Settings window appears, select **Security Settings** and press Enter.
- ix. The Primary Boot Security Settings window appears, select **Chap Name** using the up and down arrow keys and press Enter.
- x. The Enter Chap Name window appears, showing the current Chap name and an entry to put in a New Chap name. Type the new Chap name as in step a and press Enter, you return to the Primary Boot Security Settings.
- xi. On the Primary Boot Security Settings window, select Chap Secret using the up and down arrow keys and press Enter.
- xii. The Enter Old Secret window appears, type the old Chap secret and press Enter.
- xiii. The Enter New Secret window appears, type the new Chap secret as you did in step a, and press Enter.
- xiv. The Confirm New Secret window appears, type the new Chap secret again for confirmation and press Enter. You return to the Primary Boot Security Settings window.
- xv. Press Esc four times to exit. The Configuration settings modified window appears with two options: Save changes or Not to save changes. **Save changes** is already selected; press Enter.
- xvi. The Exit Fast!UTIL window appears with two options: Reboot the system or return to the Fast!UTIL utility. **Reboot System** is already selected, and press Enter.

Note: Remember that the user ID and password for CHAP are case-sensitive.

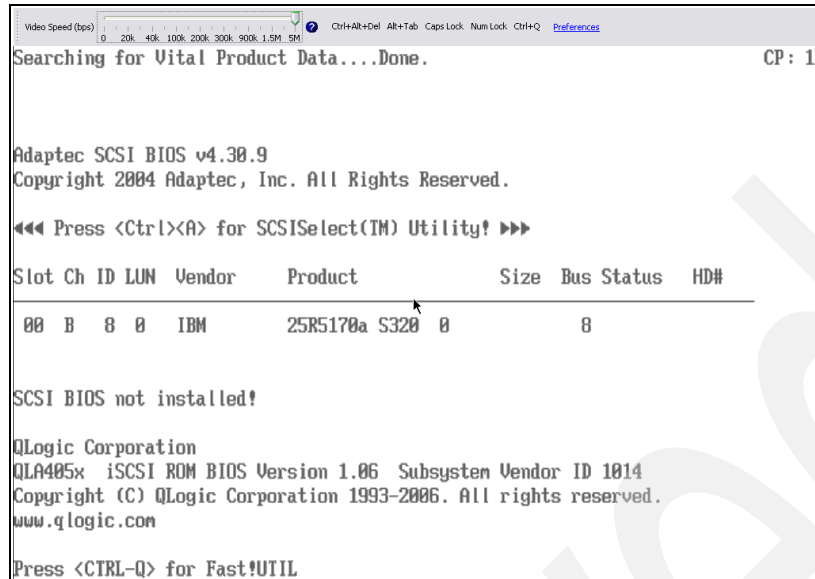


Figure 6-48 QLogic Ctrl-Q

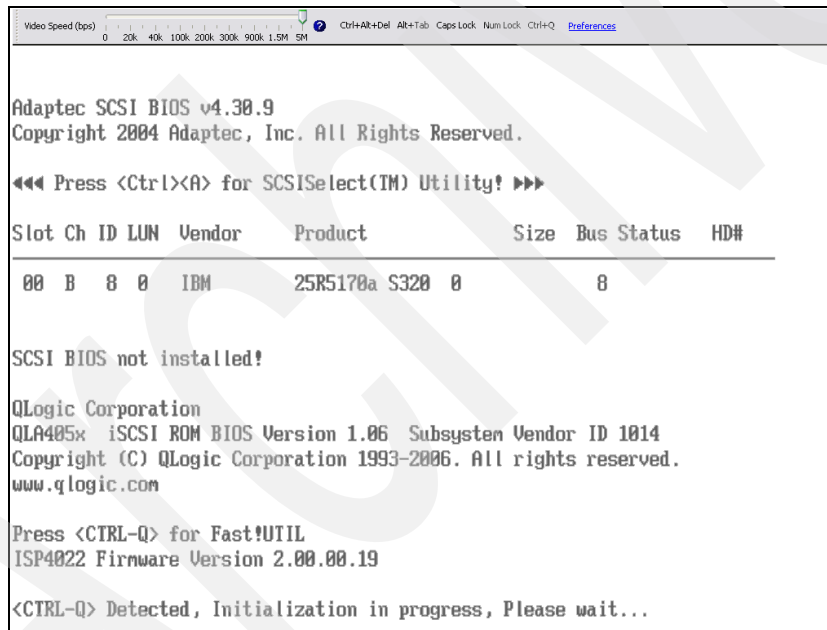


Figure 6-49 QLogic CTRL-Q Initialization

6.6.2 Manage RMTSYS object using CL command

The CL prompter can produce misleading errors when you use it to create or change a RMTSYS for multiple ports. This is a permanent restriction. So, if you want to specify more than one port, use the GUI (iSeries Navigator) to create or change the RMTSYS configuration.

► **Create a remote system configuration using CL command.**

To create a service processor configuration (NWSCFG subtype SRVPRC), follow these steps:

- a. On a 5250 command line, type CRTNWSCFG and press Enter or F4-Prompt.
- b. The Create NWS Configuration (CRTNWSCFG) display appears (Figure 6-50 on page 197).
- c. Type a **Network server configuration** (NWSCFG) name and **Configuration type** (TYPE). In this case, it is *RMTSYS for Remote System configuration, and press Enter for more parameters.
- d. Type the name of the Service Processor configuration you created earlier for **SP configuration name** (SPNWSCFG).
- e. If the RMTSYS is an xSeries type, the default value is *SPNWSCFG. If this concerns a Blade in a BladeCenter, you should type the serial number of the Blade for **Serial number** (RMTSYSID), because the *SPNWSCFG does not work. The serial number in the Service processor configuration that it references is the serial number of the BladeCenter chassis, and not the Blade itself!
- f. Type the default value *Dynamic for **Delivery Method** (DELIVERY), then the configuration is dynamically sent to the remote system using DHCP, which we recommend. The other option is *Manual, then you have to configure it manually using the CTRL-Q utility on the remote system.
- g. Type a CHAP Name for **CHAP name** (CHAPAUT), or leave the default to use the remote system configuration name instead (*NWSCFG).
- h. Leave the default *GEN for **CHAP secret** (CHAPAUT) to generate a CHAP secret for you; otherwise, type the CHAP secret, you can use special characters as well.

Important: This must match with the entry in the iSCSI adapter. The CTRL-Q utility can be used to set this. The CHAP user ID and password are case sensitive.

- i. If you have more than one iSCSI initiator on the xSeries or use the two ports on a daughter card on a Blade, you must specify which initiator does the boot. So, you must specify the **Bus, Device and Function** doing the boot for **Boot device ID** (BOOYTDEVID). These values can be found with the CTRL-Q utility during the boot of the xSeries or Blade. Press Enter for more parameters, you have page down for these.
- j. The **Vendor ID** and **Alternate ID** for **Dynamic boot options** (DYNBOOTOPT). See Figure 6-51 on page 198. There is also an option to manually configure it on the remote system, but we definitely do not recommend it. The Vendor ID is set to IBM ISAN on the initiator adapter in the xSeries or Blades, which is the only value to use. The Alternate Client ID cannot be used at the time of writing this book. Page down once again for the Remote Interface parameters, a message "At least one SCSI and LAN remote interface must be specified" appears at the bottom of the display.
- k. Type the following for the **Remote Interfaces** (RMTIFC) in Figure 6-52 on page 198:
For the **SCSI Interface**:
 - i. Type MAC-Address for **Adapter Address** of the iSCSI part of the card in the xSeries or Blade.
 - ii. Type the Internet address for **Internet Address** for the iSCSI part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH that this integrated server will use.

- iii. Type the Subnet for **Subnet mask** for the iSCSI part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH that this integrated server will use.
- iv. The **Gateway Address** can be left blank. It is not used at the time of writing of this book.
- v. Leave the default *GEN for **iSCSI qualified name** (recommended), you can specify your own. This is described in RFC3722.

For the **LAN Interface**:

- i. Type MAC-Address for **Adapter Address** of the iSCSI part of the card in the xSeries or Blade.
- ii. Type the Internet address for **Internet Address** for the iSCSI part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH that this integrated server will use.
- iii. Type the Subnet for **Subnet mask** for the iSCSI part of the card in the xSeries or Blade, which must be in the same subnet as defined in the NWSH that this integrated server will use.
- iv. The **Gateway Address** can be left blank. It is not used at the time of writing of this book.

Note: The iSCSI card has two MAC-Addresses: one for iSCSI and one for LAN connection, also called TOE.

- l. Type a Description for **Text 'description'** (TEXT) and press Enter. You see a message at the bottom of the display confirming the Remote system configuration is created.
- m. To change the **Authority (AUT)**, you have to press F10-Additional parameters.

Create NWS Configuration (CRTNWSCFG)

Type choices, press Enter.

Network server configuration . .	NWSCFG	> RMTSYS
Configuration type	TYPE	> *RMTSYS
SP configuration name	SPNWSCFG	
Remote system identifier:	RMTSYSID	
Serial number		*SPNWSCFG
Manufacturer type and model .		
Delivery method	DELIVERY	*DYNAMIC
CHAP authentication:	CHAPAUT	
CHAP name		*NWSCFG
CHAP secret		*GEN
Boot device ID:	BOOTDEVID	
Bus		> 6
Device		> 1
Function		> 1

More...

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel
 F13=How to use this display F24=More keys

Figure 6-50 First display CRTNWSCFG for RMTSYS

```

Create NWS Configuration (CRTNWSCFG)

Type choices, press Enter.

Dynamic boot options:          DYNBOOTOPT
Vendor ID . . . . .           *DFT
Alternate client ID . . . . .  *ADPT

More...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 6-51 Second display CRTNWSCFG for RMTSYS

```

Create NWS Configuration (CRTNWSCFG)

Type choices, press Enter.

Remote interfaces:          RMTIFC
SCSI interface:
Adapter address . . . . .  00C0DD0759BA
Internet address . . . . .  128.168.203.1
Subnet mask . . . . .      255.255.0.0
Gateway address . . . . .
iSCSI qualified name . . . . . *GEN

LAN interface:
Adapter address . . . . .  00C0DD0759B9
Internet address . . . . .  128.168.204.1
Subnet mask . . . . .      255.255.0.0
Gateway address . . . . .

+ for more values
Text 'description' . . . . . TEXT      Remote System Configuration

Bottom
F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys
At least one SCSI and LAN remote interface must be specified.  +

```

Figure 6-52 Third display for CRTNWSCFG for RMTSYS

► **Display remote system configuration using CL command.**

To display a Remote system configuration, follow the steps:

- a. On a 5250 command line, type DSPNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG(*Remote system configuration name*) and put option 5-Display on the option column in front of the SRVPRC to display.
- b. The Display NWS Configuration (DSPNWSCFG) display appears:
- c. Type the name of the SRVPRC, and press Enter.
- d. Two more parameters appear, **Option** and **Output**.
 - v. On the **Option** parameter, you can type *ALL, *BASIC, *RMTSYS, or *RMTIFC for a RMTSYS configuration. If you type *ALL, you see all options. On the *RMTSYS option, you can page down to see the other parameters.
 - i. On the **Output** parameter, you can type * to display it on the window or *Print to create a spoolfile.

► **Change remote system configuration using CL command.**

To change a Remote System configuration, follow the steps:

- On a 5250 command line, type CHGNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG (Remote system configuration name) and put option 2-Change on the option column in front of the RMTSYS to change.
- The Change NWS Configuration (CHGNWSCFG) display appears (Figure 6-53 and Figure 6-54 on page 200).
- Type the name of the **RMTSYS**, and press Enter.
- The first three parameters are shown, press Enter again to see the rest and page down to the Remote Interface configuration. Press Enter again to make the changes.

Change NWS Configuration (CHGNWSCFG)			
Type choices, press Enter.			
Network server configuration . . .	> RMTSYS	Name	
SP configuration name	MMSP	Name, *SAME	
Remote system identifier:			
Serial number	*SPNWSCFG	Character value, *SAME...	
Manufacturer type and model .		Character value	
Delivery method	*DYNAMIC		
CHAP authentication:			
CHAP name	'RMTSYS'		
CHAP secret	'1hH;uIcC(JrCxip=QPYwQKHr'		
Boot device ID:			
Bus	*SINGLE	0-255, *SAME, *SINGLE	
Device		0-31	
Function		0-7	
Dynamic boot options:			
Vendor ID	*DFT	Character value, *SAME, *DFT	
Alternate client ID	*ADPT	Character value, *ADPT	
More...			
F3=Exit	F4=Prompt	F5=Refresh	F12=Cancel
F24=More keys	F13=How to use this display		

Figure 6-53 First display of CHGNWSCFG for RMTSYS configuration

```

Change NWS Configuration (CHGNWSCFG)

Type choices, press Enter.

Remote interfaces:
  SCSI interface:
    Adapter address . . . . . 02000000000F Hexadecimal value, *SAME...
    Internet address . . . . . '128.168.203.1'
    Subnet mask . . . . . '255.255.0.0'
    Gateway address . . . . .
    iSCSI qualified name . . . . . '*GEN'
  LAN interface:
    Adapter address . . . . . 02000000000E Hexadecimal value, *NONE
    Internet address . . . . . '128.168.204.1'
    Subnet mask . . . . . '255.255.0.0'
    Gateway address . . . . .
      + for more values
  Text 'description' . . . . . 'Remote System Configuration'

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
Bottom

```

Figure 6-54 Second display for CHGNWSCFG for RMTSYS configuration

► **Display remote system status using CL command.**

The hardware status of the xSeries or BladeCenter server can be retrieved. This comes in handy to check if IBM Director server is able to reach and retrieve the status of the server.

When it fails to retrieve information, you notice that option 8-Work with status on the Work with NWS Configuration display fails with a message in your joblog. You can press F1 on the message at the bottom of the window or type DSPJOBLOG on the command line. The hardware status could be:

- Offline
- Vary on pending/POST
- Booting (in POST)
- Failed
- Vary on pending/BIOS
- Vary on pending/OS
- Active
- Vary on Pending/CPU reset
- Vary on Pending/Power on
- Vary on Pending/Powered on

You can also monitor the status during the vary on of the iSCSI integrated server by pressing F5-Refresh on the status display so that you can follow the state of the server.

To display the status of a Remote system configuration, follow the steps:

- a. On a 5250 command line, type WRKNWSCFG and press F4-prompt, type *RMTSYS for **Option**. You can leave the default *ALL but option 8-Work with status is only valid with the RMTSYS configuration. Another option is to use the DSPNWSCFG command.
- b. The Work with NWS Configuration (WRKNWSCFG) display appears.
- c. Scroll down to the RMTSYS for which you want to display the status, and type 8-Work with status in front of RMTSYS configuration.

- d. The Work with Status display appears, the status is shown. You can press F5-Refresh to refresh the status. During a boot, you can follow the status.

► **Delete a remote system configuration using CL command.**

To delete a Remote system configuration, follow the steps:

- a. On a 5250 command line, type WRKNWSCFG and press F4-prompt, type *RMTSYS for **Option**. You can leave the default *ALL. Another option is to use the DLTNWSCFG command.
- b. The Work with NWS Configuration (WRKNWSCFG) display appears.
- c. Scroll down to the RMTSYS you want to delete and type 4-Delete on the option column in front of RMTSYS configuration.
- d. The Confirm Delete of NWS Configuration display appears, press Enter. You see a status message showing the configuration is deleted.

► **Change the CHAP authentication using CL command.**

The CHAP authentication can be deferred from the remote system configuration by checking the *RMTSYS option of doing a display of remote system configuration with the command DSPNWSCFG or WRKNWSCFG. If you want to change the CHAP authentication, this has to be done on two places: 1. Within the Remote System configuration and 2. Within the CTRL-Q utility on the xSeries server or a Blade within the BladeCenter chassis to configure the iSCSI card.

To change the CHAP authentication, follow the steps:

- a. Change the remote system configuration's CHAP as described in Change remote system configuration properties using CL command with the new values for the CHAP name and CHAP secret for parameters **CHAP Name** and **CHAP Secret** (CHAPAUT).
- b. Start the CTRL-Q utility on the xSeries server or the correct Blade in a BladeCenter. You can do this on the console of the xSeries or BladeCenter or within the remote console utility using a WEB browser pointing to the service processor of the xSeries or BladeCenter described in 6.8.6, "Starting Remote Control on MM to see Windows booting" on page 229 for an xSeries and in 6.8.7, "Starting Remote Control on RSA II to see Windows booting" on page 230 for a BladeCenter.

Note: For BladeCenter, select first the right Blade for the Keyboard, Video, and Mouse (KVM) to use the monitor, keyboard, and mouse by selecting the monitor icon at the left top of the Blade.

- i. Start the xSeries or Blade with the power on button or within the WEB browser pointing to the service processor. If it is already up and running, you can also restart the server, so POST (Power-On Self Test) will start on the server.
- ii. Some time after the eServer logo displays, the QLogic bios prompt shows. It reads "Press <CTRL-Q> for Fast!UTIL." At this time, issue CTRL-Q. If you are using the remote console utility, select **Ctrl** at the toolbar with the MM of the Blade center. For the xSeries, you must configure this first using preferences.
- iii. When this is successful, it reads "<CTRLQ> Detected, Initialization in progress, Please wait". The QLogic Fast!UTIL menu appears.
- iv. If more than one adapter is installed, select the one to change the CHAP authentication using the up and down arrow keys, and press Enter.
- v. The selected adapter is shown at the top and on another pane, Fast!UTIL options, select **Configuration Options**, and press Enter.

- vi. The Configuration Options window appears, select **iSCSI Boot settings** using the up and down arrow keys, and press Enter.
- vii. The iSCSI Boot settings window appears, select **Primary Boot Device Settings** using the up and down arrow keys, and press Enter.
- viii. The Primary Boot Device Settings window appears, select **Security Settings**, and press Enter.
- ix. The Primary Boot Security Settings window appears, select **Chap Name** using the up and down arrow keys and press Enter.
- x. The Enter Chap Name window appears, showing the current Chap name and an entry to put in a **New** Chap name. Type the new Chap name as you did in step a, and press Enter. You return to the Primary Boot Security Settings window.
- xi. On the Primary Boot Security Settings window, select **Chap Secret** using the up and down arrow keys, and press Enter.
- xii. The Enter Old Secret window appears, type the old Chap secret and press Enter.
- xiii. The Enter New Secret window appears, type the new Chap secret as you did in step a, and press Enter.
- xiv. The Confirm New Secret window appears, type the new Chap secret again for confirmation and press Enter. You return to the Primary Boot Security Settings window.
- xv. Select <ESC> to exit four times. The Configuration settings modified window appears with two options: Save changes or Not to save changes. Save changes is already selected, press Enter.
- xvi. The Exit Fast!UTIL window appears with two options: Reboot the system or return to the Fast!UTIL utility. Reboot System is already selected, press Enter.

Note: Remember that the user ID and password for CHAP are case-sensitive.

6.7 Manage connection security configurations

As we have described before, the connection security network server configurations are not used at the time of writing this redbook; however, it needs to exist to make the iSCSI connection to work.

6.7.1 Manage connection security configuration using iSeries Navigator

- **Create a connection security configuration using iSeries Navigator.**

To create a CNNSEC configuration, follow the steps (Figure 6-55 on page 203):

- a. Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- b. Right-click **Connection Security** and select **New Connection Security Configuration**.
- c. The New Connection Security Configuration window opens. On the General tab, enter the following:
 - i. **CNNSEC Name**
 - ii. **CNNSEC Description**
- d. Select **Object authority** and click **OK**.

Note: The IP Security Rules page cannot be filled at the time of writing this book. There is no reason to select it during the creation at this time.

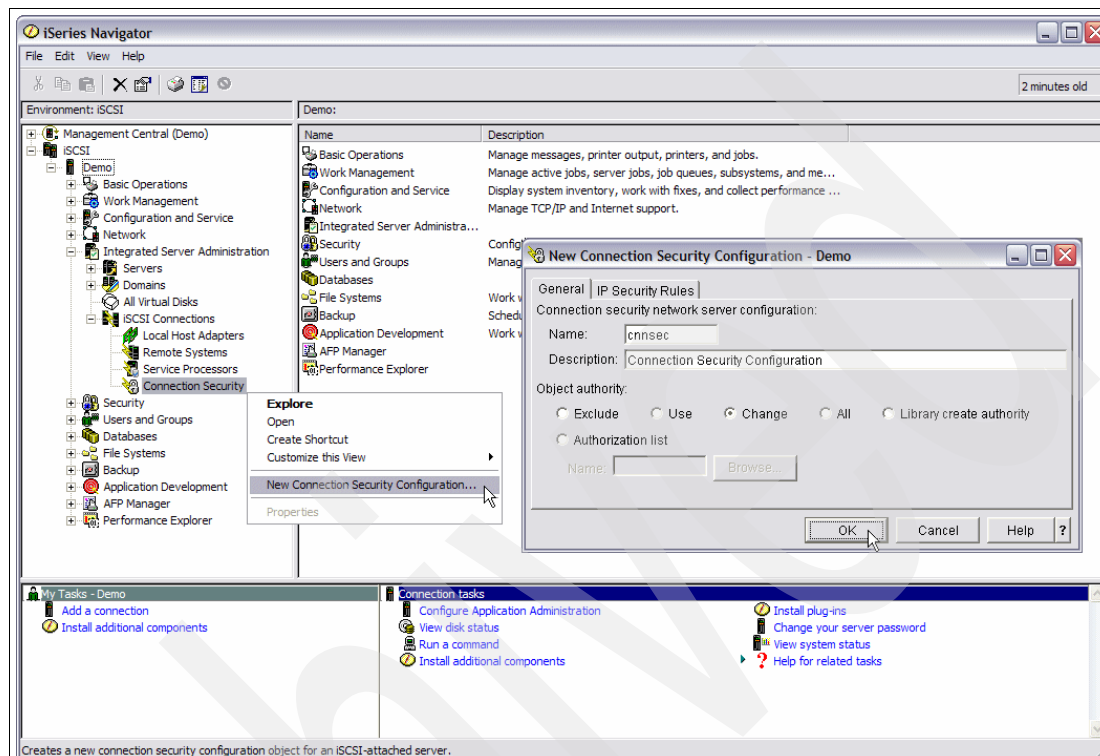


Figure 6-55 New Connection Security Configuration in iSeries Navigator

► **Create a connection security configuration based on another one using iSeries Navigator.**

To create a CNNSEC configuration based on another one, follow the steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**.
- Click **Connection Security** and right-click the **Connection Security** configuration at the right pane which you want to be copied from, and select **New Based On**.
- Enter a new CNNSEC name and change any other attribute as described in Create a connection security configuration using iSeries Navigator and click **OK**.

► **Display/Change a connection security configuration using iSeries Navigator.**

To display or change a CNNSEC configuration, you follow the same steps. The only attribute you can change is the Description on the general tab:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**. See Figure 6-56 on page 204.
- Click **Connection Security** and right-click the **Connection Security** configuration at the right pane which you want to display or change from, and select **Properties**.

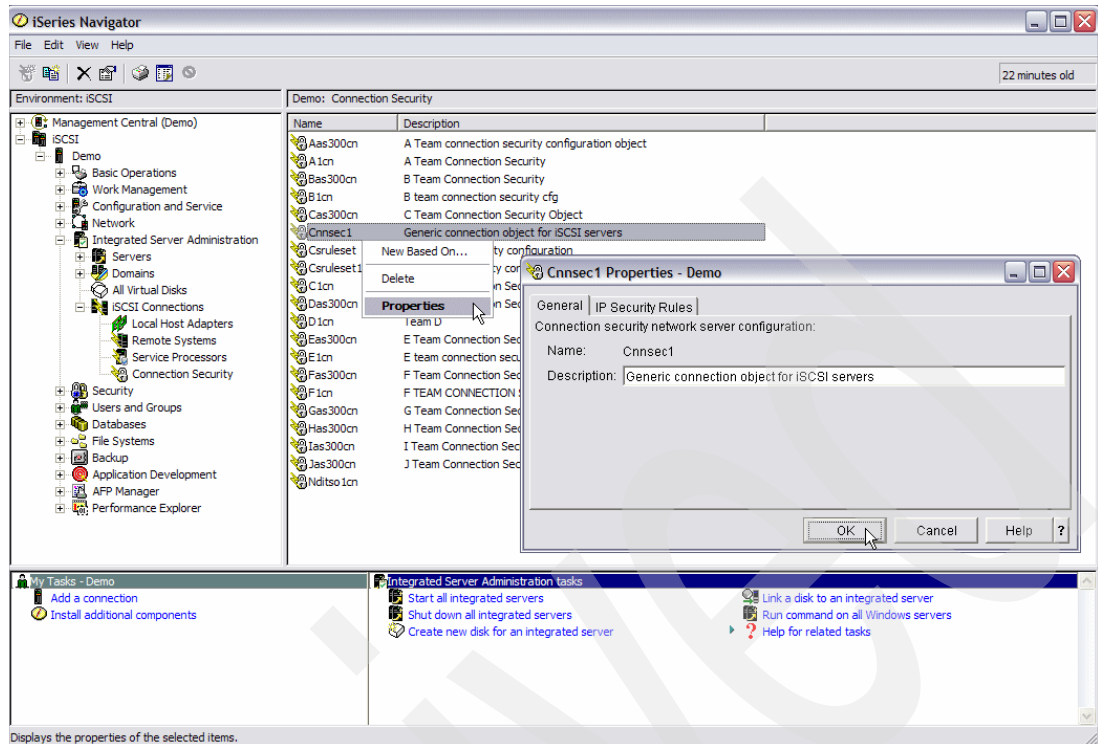


Figure 6-56 Properties CNNSEC in iSeries Navigator

► **Delete a connection security configuration using iSeries Navigator.**

To delete a CNNSEC configuration, you follow these steps:

- Expand **YourSystem** → **Integrated Server Administration** → **iSCSI Connections**. See Figure 6-57 on page 205.
- Click **Connection Security** and right-click the **Connection Security** configuration at the right pane which you want to delete from, and select **Delete**.
- The Confirm Delete Connection Security Configuration window opens, click **Delete**.

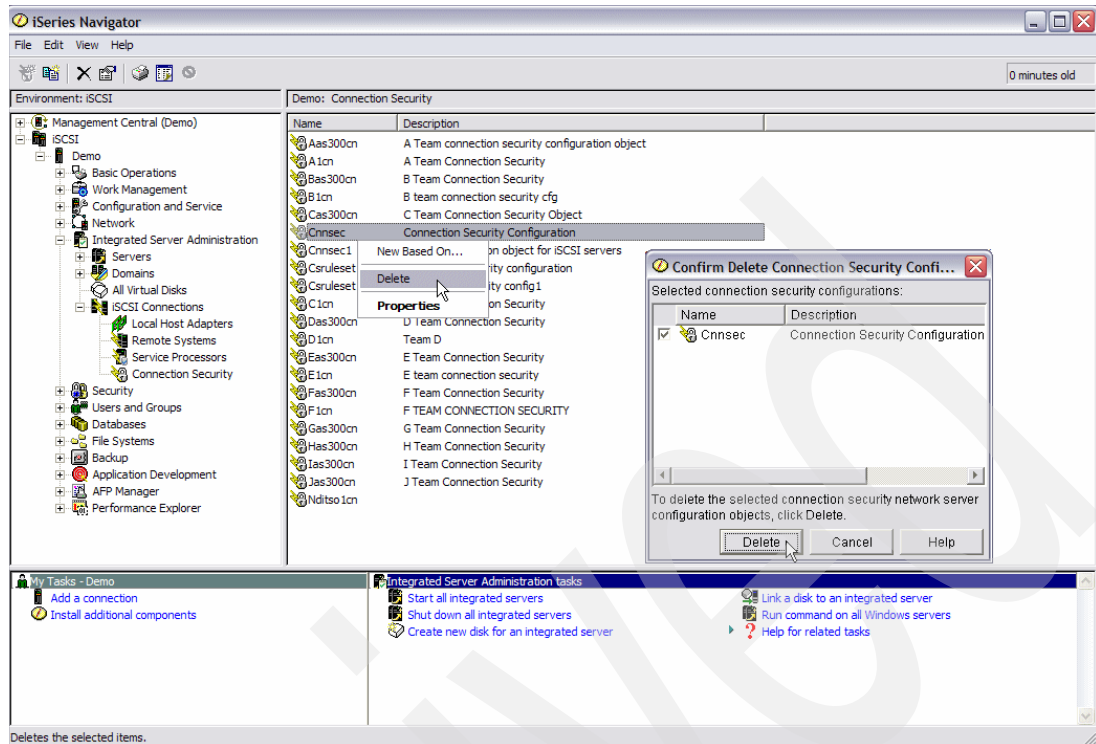


Figure 6-57 Delete CNNSEC in iSeries navigator

6.7.2 Manage connection security configuration using CL command

► Create a connection security configuration using CL command.

To create a connection security configuration (NWSCFG subtype CNNSEC), follow these steps:

- On a 5250 command line, type CRTNWSCFG and press Enter or F4-prompt.
- The Create NWS Configuration (CRTNWSCFG) display appears (Figure 6-58 on page 206).
- Type a **Network server configuration** (NWSCFG) name and **Configuration type** (TYPE). In this case, it is *CNNSEC for Connection Security configuration, and press Enter for more parameters.
- Type *NONE for IP Security rules (IPSECRULE). This is the only option supported at this time. Support might be provided at a future date.
- Type a Description for **Text 'description'** (TEXT), and press Enter. You see a message at the bottom of the display confirming the Network server configuration is created.
- To change the **Authority** (AUT), you have to press F10-Additional parameters.

Create NWS Configuration (CRTNWSCFG)

Type choices, press Enter.

Network server configuration . . . >	CNNSEC	Name
Configuration type >	*CNNSEC	*CNNSEC, *RMTSYS, *SRVPRC
IP security rules	*NONE	
	+ for more values	
Text 'description' >	'Connection Security Configuration'	

Additional Parameters

Authority	*CHANGE	Name, *CHANGE, *ALL, *USE...
---------------------	---------	------------------------------

Bottom

F3=Exit	F4=Prompt	F5=Refresh	F12=Cancel	F13=How to use this display
F24=More keys				

Figure 6-58 CRTNWSCFG for CNNSEC

► **Display connection security configuration using CL command.**

To display a Service processor configuration, follow the steps:

- a. On a 5250 command line, type DSPNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG(Connection security name) and put option 5 (Display) on the option column in front of the CNNSEC to display.
- b. The Display NWS Configuration (DSPNWSCFG) display appears.
- c. Type the name of the CNNSEC, and press Enter.
- d. Two more parameters appear, **Option** and **Output**.
 - iii. On the Option parameter, you can type *ALL, *BASIC or *CNNSEC for a CNSEC configuration. If you type *ALL, you see all options.
 - i. On the Output parameter, you can type * to display it on the display or *Print to create a spoolfile.

► **Change connection security configuration using CL command.**

To change a Connection Security configuration, follow the steps:

- a. On a 5250 command line, type CHGNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG (Connection Security name) and put option 2 (Change) on the option column in front of the CNNSEC to change.
- b. The Change NWS Configuration (CHGNWSCFG) display appears (Figure 6-59 on page 207).
- c. Type the name of the CNNSEC, and press Enter.
- d. The only parameter you can change is **Text 'description'**.


```

Change NWS Configuration (CHGNWSCFG)

Type choices, press Enter.

Network server configuration . . > CNNSEC      Name
IP security rules . . . . . *NONE
      + for more values
Text 'description' . . . . . 'Connection Security Configuration'

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 6-59 CHGNWSCFG for CNNSEC

► **Delete connection security configuration CL command.**

To delete a Connection Security configuration, follow the steps:

- On a 5250 command line, type DLTNWSCFG and press Enter or F4-prompt. Another option is to type WRKNWSCFG NWSCFG (Connection Security name) and put option 4 (Delete) on the option column in front of the CNNSEC to delete.
- The Delete NWS Configuration (DLTNWSCFG) display appears (Figure 6-60).
- Type the name of the CNNSEC, and press Enter. A message appears that the object is deleted.

```

Delete NWS Configuration (DLTNWSCFG)

Type choices, press Enter.

Network server configuration . . CNNSEC      Name

                                                                    Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys

```

Figure 6-60 DLTNWSCFG for CNNSEC

6.7.3 Manage network server description (NWSD) using iSeries Navigator

The Network Server Description is different for an iSCSI integrated server than it is for an IXS or IXA. The NWSD is still important, only it has a different function. The NWSD object is used to tie together the other i5/OS objects that relate to an iSCSI integrated server. Table 6-1 on page 208 shows the tabs for an IXS/IXA server in comparison to an iSCSI NWSD. As you see, there is quite a difference. In the Display/Change Network Server Description, we describe the contents of each tab for iSCSI.

Table 6-1 NWSD tabs comparison for IXS/IXA and iSCSI

IXS/IXA NWSD Tabs	iSCSI NWSD Tabs
General	General
OS/400	System
Software	Software
Messages	Messages
Disk Drives	Storage Paths
	iSCSI Security
	TCP/IP
	Virtual Ethernet

► **Display/Change Network Server Description using iSeries Navigator.**

When the NWSD has the status of started or starting, you are limited to change attributes and you are not able to add storage paths or Virtual Ethernets. The only attributes, which can be changed in this state are:

On the System tab Description.

Advanced Button: Synchronize date and time.

Advanced Button: Propagate Domain Users.

On the Messages tab The message you want to receive from windows, which are System, Security, and Application.

To display or change a NWSD configuration, you follow the same steps:

- Expand **YourSystem** → **Integrated Server Administration** → **Servers** or click **Servers**.
- Right-click the Server you want to display or change, and select **Properties**. The Properties window opens.

► **General Tab Information for general information of this integrated server (Figure 6-61 on page 209):**

- *Server Name*: The integrated server computer name
- *Description*: The description of the NWSD
- *Server Role*: Role of the server in the Windows Domain (Domain Server or Server)
- *Domain*: The Netbios name of the Windows domain this server belongs to
- *DNS domain name*: The DNS domain name of this server

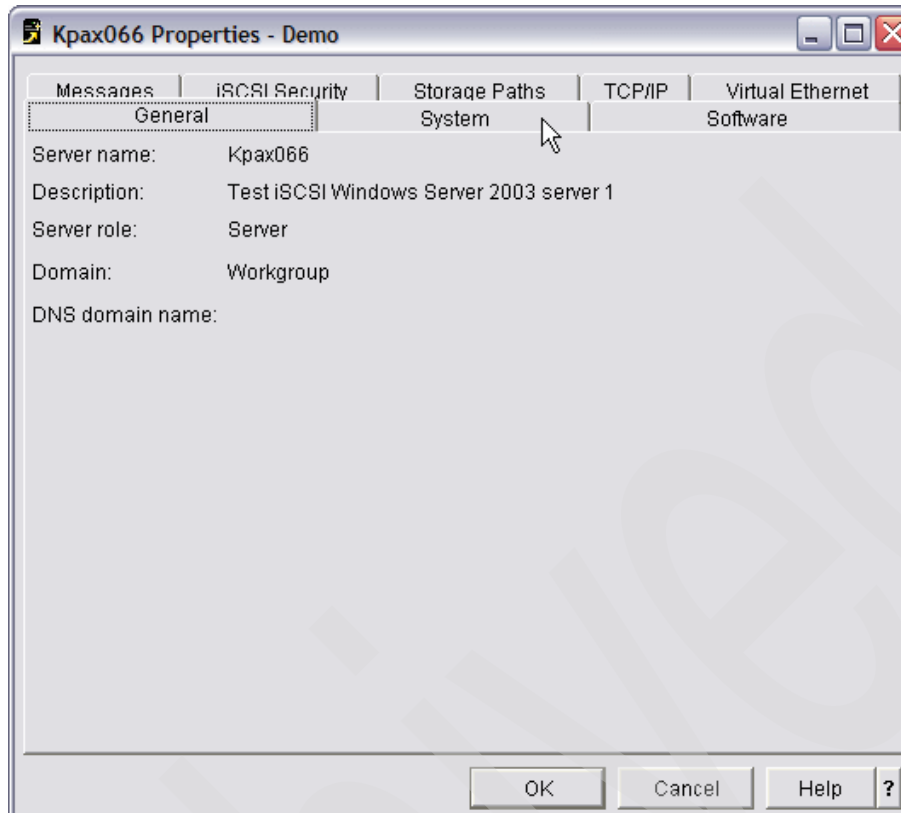


Figure 6-61 General tab NWSD Properties

Click the **System** tab of the properties window of the NWSD (Figure 6-62 on page 210):

System page information for system information, such as type of server connection which RMTSYS configuration uses:

- ▶ **Name:** The NWSD name
- ▶ **Description:** The Description of the NWSD object
- ▶ **Type of server connection:** For which the NWSD is used
 - iSCSI for NWSD type *iSCSI used for iSCSI attached xSeries or Blades
 - Integrated for NWSD type *IXSVR for an IXS or IXA
 - Logical Partition for NWSD type *GUEST for Linux and AIX running in a partition
- ▶ **Type of server operating system:** Type of operating system installed, such as Windows
- ▶ **Hardware pane:**
 - **Remote system configuration name:** Name of the RMTSYS configuration used
 - **Remote system type:** The type and model the RMTSYS points to
 - **Type at installation:** The type and model used at installation time
- ▶ **Properties button:** Opens the RMTSYS properties window to display or change

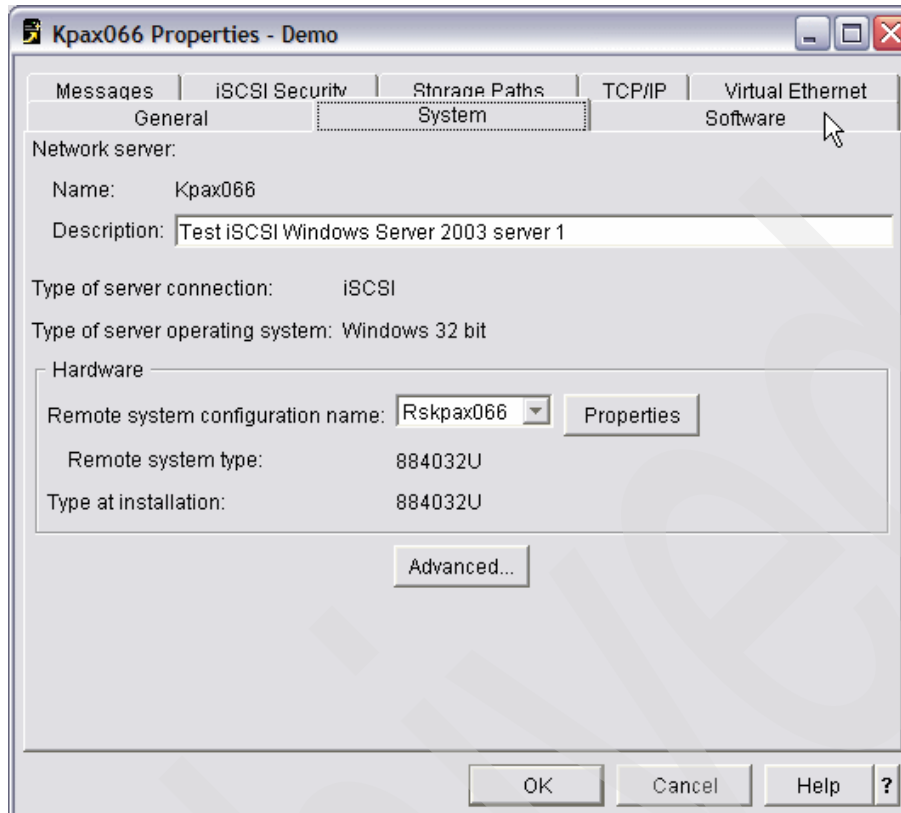


Figure 6-62 System tab NWSD Properties

Click **Advanced** on the System page, you can set configuration settings, such as synchronize date and time and to restrict devices. There are two tabs: Configuration and Restricted Devices:

► **Configuration page information (Figure 6-63 on page 211):**

- **Synchronize date and time:** Select this box to synchronize date and time with i5/OS.
- **Propagate domain users:** This parameter is important for Windows user enrollment. If you have multiple Windows servers on your iSeries, you should select only one server, usually a domain controller, to perform the user enrollment. Otherwise, each server on the iSeries propagates users as they are varied on. Check the box if this is the server you want to use for user enrollment.
- **Language version:** The language of the iSeries integrated server support product.
- **Code page:** Code page used for ASCII text files on the server.
- **Use configuration file:** Select this option to specify a source file to activate or further define the server, a file and library should be specified.
- **Hardware shut down timeout:** Specifies the amount of time in minutes before the server hardware is forced offline. The default for Windows is 15 minutes for the time to normally shut down. If this is a heavily used server and it needs more time for the shutdown, increase as needed.
- Advanced iSCSI options pane:
 - **Shutdown TCP port:** The TCP port number that is used to signal the server to shut down. This port is on the xSeries or Blade (8700 is default).
 - **Virtual Ethernet control port:** The TCP port number that is used to control Virtual Ethernet. This port is on the xSeries or Blade (8800 is default).

- **Activation timer:** The time the iSeries waits to establish a connection to the service processor of the xSeries or BladeCenter chassis (120 seconds is default).

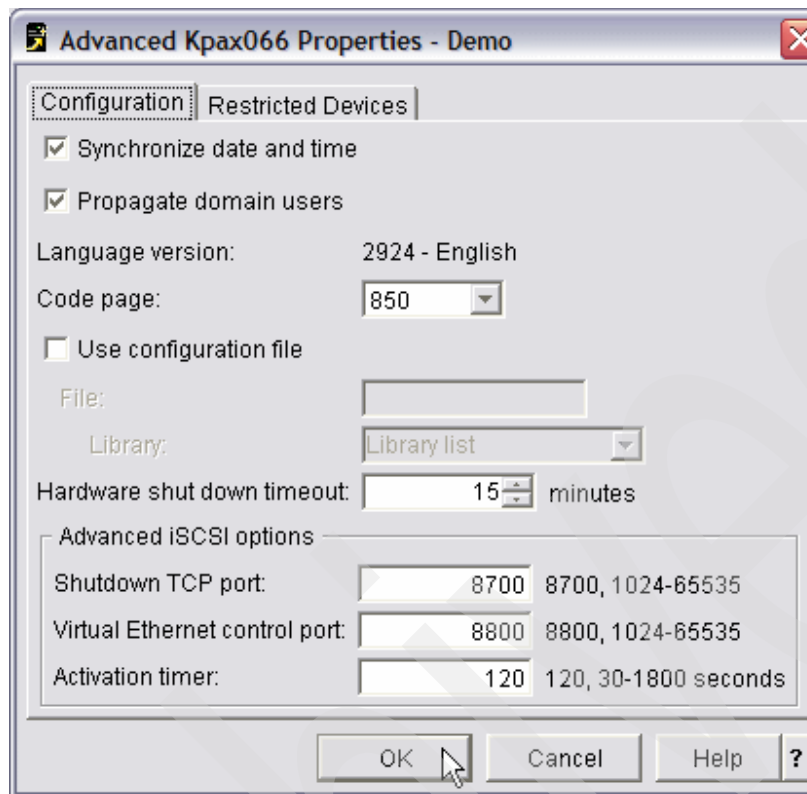


Figure 6-63 Configuration tab of the Advanced button on System tab

Click the **Restricted Devices** tab on the Advanced properties window for restricted devices settings:

► **Restricted Devices page information (Figure 6-64):**

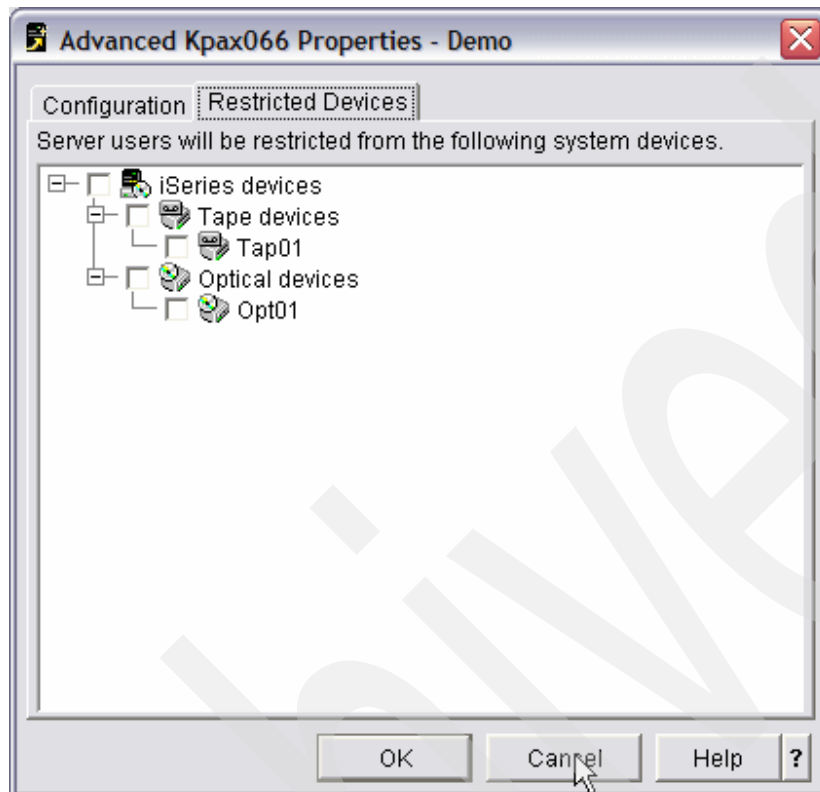


Figure 6-64 Restricted Devices tab of the Advanced button on System tab

- Select the devices, which are restricted from use by the integrated server:
 - **iSeries devices:** Select this box to restrict all tape and optical devices from use by the integrated server.
 - **Tape devices:** Select this box to restrict all tape devices from use by the integrated server. You are also able to restrict specific tape resources by clicking the box for the tape resource under tape devices.
 - **Optical devices:** Select this box to restrict all optical devices from use by the integrated server. You are also able to restrict specific optical resources by clicking the box for the optical resource under optical devices.

Click **Cancel** on the Restricted Devices page of the Advanced properties window and click the **Software** tab of the properties of the NWSD window.

► **Software page information for information about software levels (Figure 6-65 on page 213):**

- Operating system (OS) version pane: Information of the integrated server.
 - Version: The version of the Windows operating system installed
 - Build ID: The current build level of the Windows operating system
 - Service level: The service pack level of the Windows operating system
- iSeries Integrated Server Support Version: Information of the i5/OS integrated Server support product (5722-SS1 option 29):

- Version: Release of the integrated server support product
- Language version: Language of the integrated server support product
- Service pack: The PTF installed for the integrated server support product on Windows.

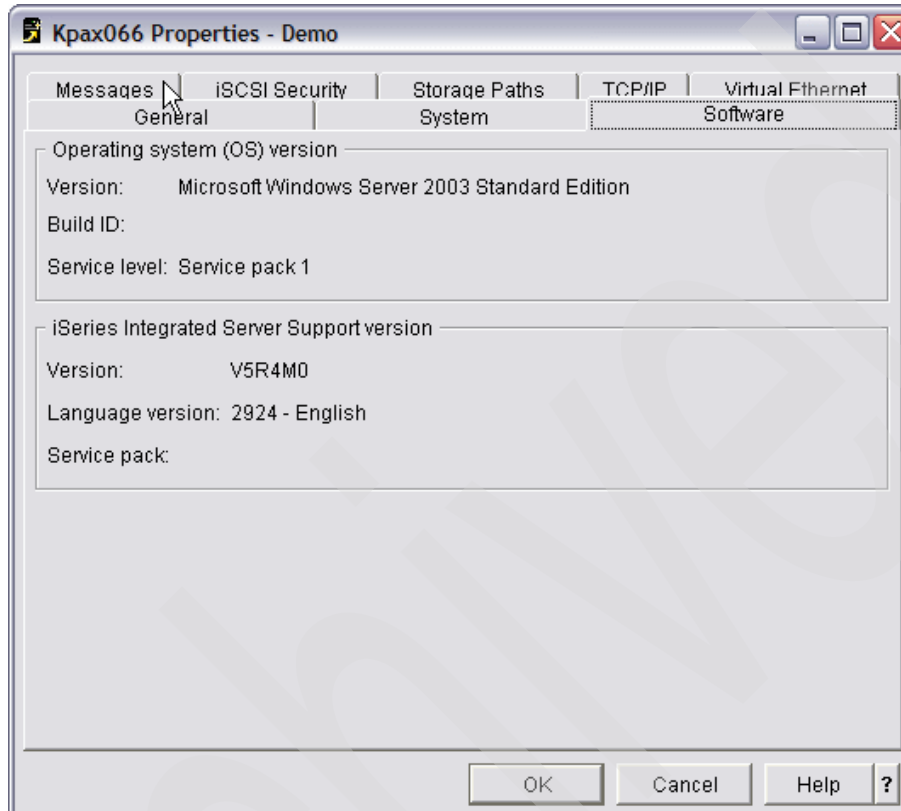


Figure 6-65 Software tab NWSD properties

Click the **Messages** tab to see the way message logging is set for events from Windows:

► **Messages page information (Figure 6-66 on page 214):**

- Server message logging pane to select where the event messages from Windows should be reported:
 - Do not log messages: Select this radio button if you do not want to log event messages from the Windows event log to the i5/OS.
 - Log Messages to job log: Select this radio button if you want to log the event messages from the Windows event log to the joblog of the job with the same name as the NWSD in subsystem QSYSWRK.
 - Log Messages to server message queue: Select this radio button if you want to log the event message from the Windows event log to the specified message queue name and library. Message queue must exist. You can also click Browse to search for one.
- Messages to receive from the Windows event log pane to select which event messages should be logged.
 - System messages: Select this box to receive the System messages from the Windows event log.

- Security messages: Select this box to receive the Security messages from the Windows event log.
- Application messages: Select this box to receive the Application messages from the Windows event log.
- Communications message logging: Specify an existing message queue and library or browse for one by clicking Browse for iSCSI communications status messages.

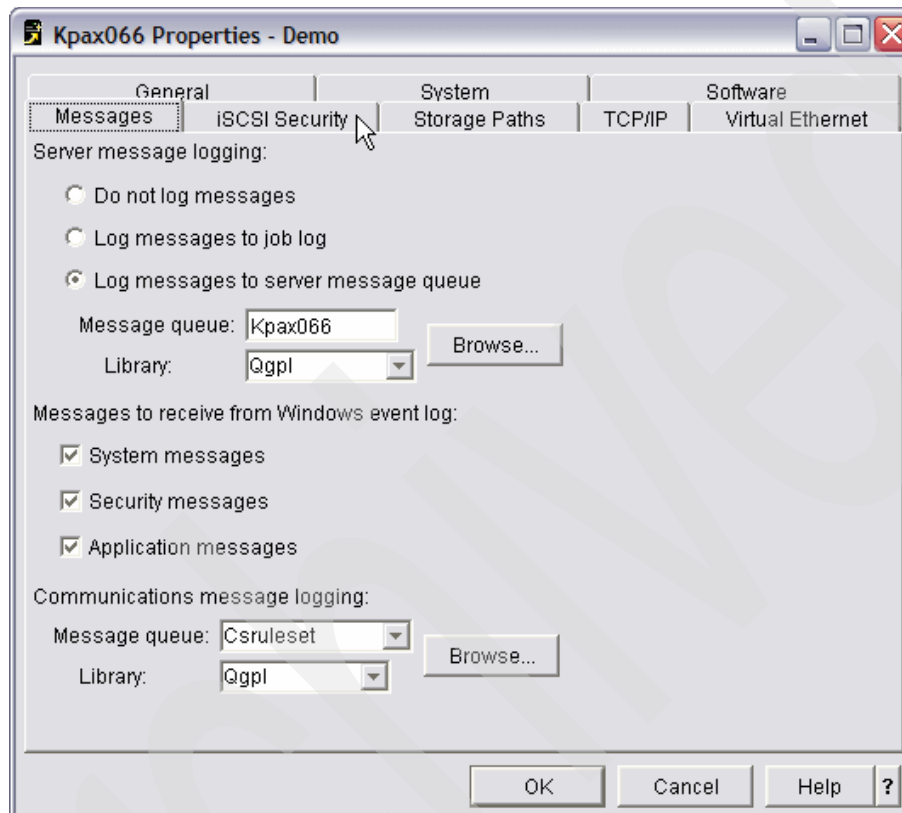


Figure 6-66 Messages tab NWSD properties

Click the **iSCSI Security** tab to see which Connection Security configuration is used.

► **iSCSI Security page information (Figure 6-67 on page 215):**

- Connection security configuration: The name of the connection security configuration used by this NWSD. Use the pull-down list to see connection security configurations. You can click the Properties button to see the properties of this configuration.
- Default IP security rule: At the time of writing this book, None is the only option.

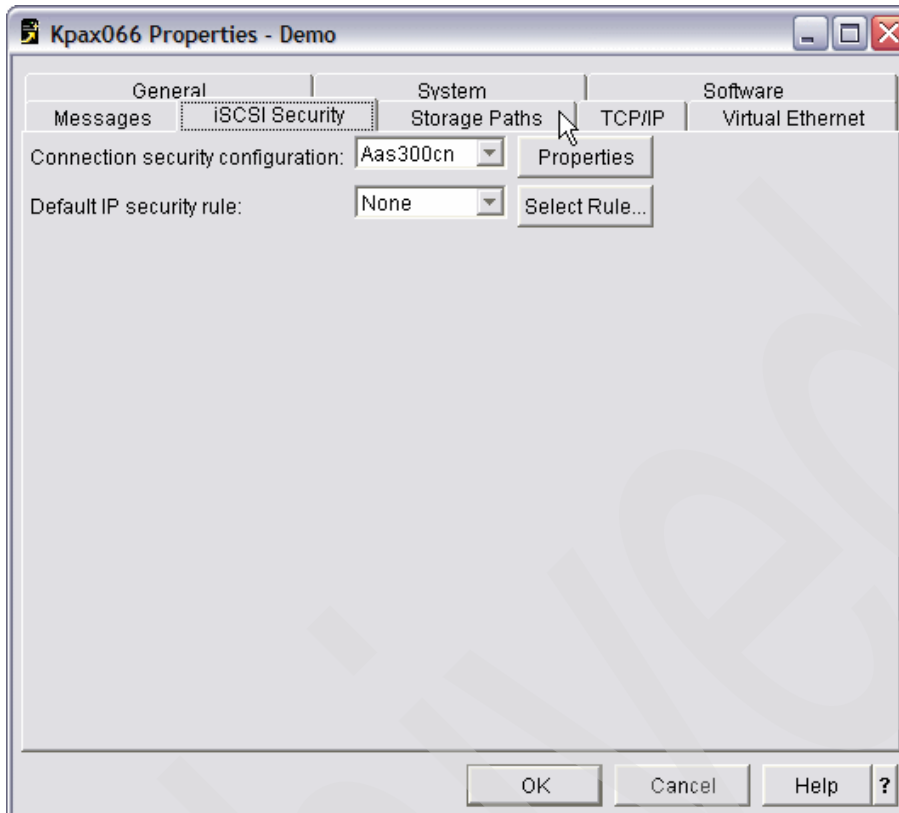


Figure 6-67 iSCSI Security tab NWSD properties

Click the **Storage Paths** tab for storage paths configurations used by this NWSD. You can create, check the properties, remove, and change the order by using the buttons Add, Properties, Remove, Move UP, and Move Down at the right side of the box (Figure 6-68 on page 216).

- ▶ **Path used to access storage devices:** The box shows the storage paths defined for the integrated server in this NWSD. It also shows how the Host Bus Adapter (HBA) is used on the System i5. The maximum HBAs in System i5 are eight.

- The columns shown are:

- **Path:** Indicates the storage path number. This is the number that is used to identify this path.
- **NWSH Name:** The name of the network server host adapter (NWSH) object that the path uses.
- **Resource:** The name of the resource that is being used for the storage path. If the network server description (NWSD) is not varied on or never varied on, then *None* is shown.
- **Resource Status:** The status of the file server resource. One of the following is shown:

No status is shown if the resource name is *None*. Status NWSD is varied off.

Varied off: Resource or NWSH is varied off.

Ready: Resource is ready to use, but not actively being used.

Active: Resource is actively being used.

Hard failure: Resource has a hard failure.

Soft failure: Resource has a soft failure.

- iSCSI Qualified Name (IQN): The IQN that is used for the storage path of the HBA in the i5/OS.
- Remote Interface IP Security Rule: At the time of writing this book, this is not supported.
- Multi-path group is not supported at the time of writing this book.
- Default path for disk drives: Select the default storage path used when linking storage spaces selected from the pull-down list of the defined NWSH list.
- Removable media path: Select the storage path to use for tape and optical selected from the pull-down list of the defined NWSH list.

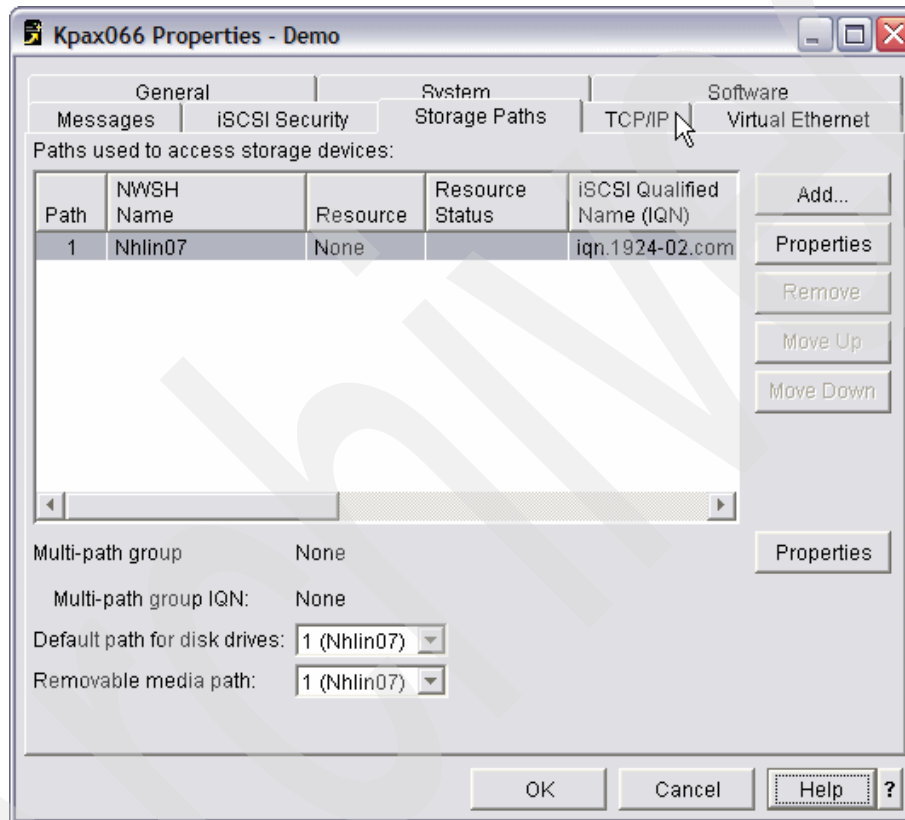


Figure 6-68 Storage Paths tab NWSD Properties

Click the **TCP/IP** tab for the TCP/IP settings:

► **TCP/IP page information (Figure 6-69 on page 217):**

- Local host name: The TCP/IP host name for this integrated server. Default is the same name as the NWSD.
- Local domain name: The TCP/IP domain name for this integrated server. Default is the same domain as specified in the TCP/IP domain information in i5/OS.

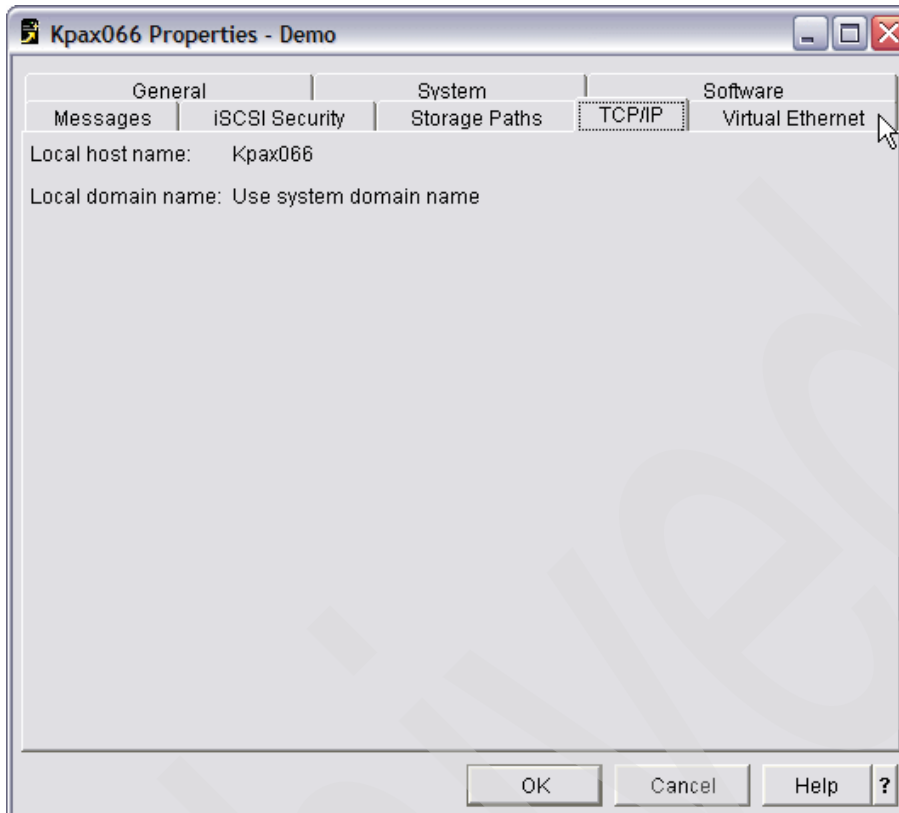


Figure 6-69 TCP/IP tab NWSD Properties

Click the **Virtual Ethernet** tab. The box shows the Virtual Ethernet ports defined for the integrated server in this NWSD. This includes the Virtual Ethernet Point-to-Point as well. The maximum different Virtual Ethernet ports defined is eight. You can create, see properties, or remove Ethernet ports using the buttons Add, Properties, and Remove at the right side of the box (Figure 6-70 on page 218).

- ▶ **Configured Virtual Ethernet ports:** The box can show up to eight defined Ethernet ports for this NWSD.
 - The columns shown are:
 - **Virtual Ethernet:** Indicates the Virtual Ethernet port that is defined. One of the following is shown:
 - n: The Virtual Ethernet port number (0-9).
 - Point-to-point: The Virtual Ethernet Point-to-Point private LAN port.
 - **Internet Address:** The Internet address for the integrated server side of the Virtual Ethernet port.
 - **Subnet Mask:** The subnet mask for the Virtual Ethernet port.
 - **Line Description:** The line description that is used for the Virtual Ethernet port.
 - **NWSH Name:** The name of the network server host adapter (NWSH) object for the path that the port uses.
 - **Resource:** The name of the Virtual Ethernet resource for the path.
 - **Remote Interface IP Security Rule:** At the time of writing this book, this is not supported.

- Virtual Ethernet LAN participation: The box shows which NWSDs are using which ports for Virtual Ethernet. Point-to-Point and Stand-alone logical partitions (LPARs), including the hosting LPAR, are not shown. Click **Cancel**.

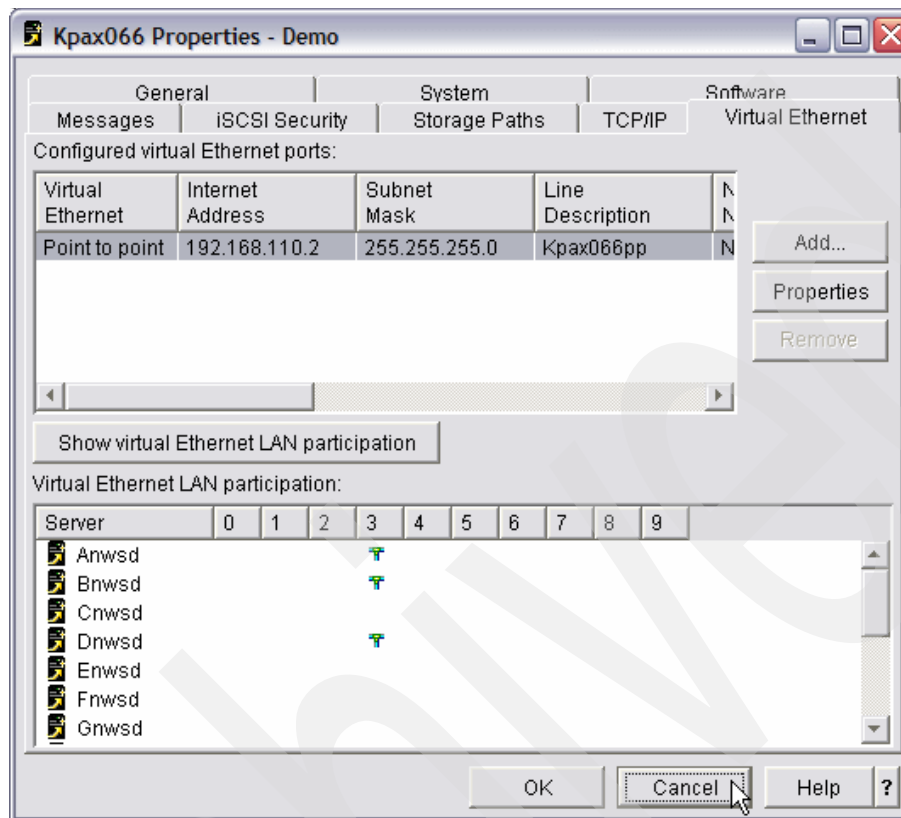


Figure 6-70 Virtual Ethernet tab NWSD Properties

6.8 Starting iSCSI integrated server

The INSWNTSVR command, which installs the Windows server also for iSCSI, does not configure the newly created server to autostart. There are several ways to start the server. You can configure it to autostart, let a CL program start the iSCSI connection, or do it manually.

There is a specific change with iSCSI connections and the way you can activate or deactivate the iSCSI integration server. For instance, you are able to use shutdown from Windows, start the xSeries or Blade server by pressing the white power button, or start it using the WEB browser pointing to the service processor and watch the boot process using Remote control. To be able to use the power button or remote control, the NWSD should have the status of VARIED ON. This NWSD gets to this status when doing a shutdown from Windows or the remote control function.

If the NWSH device does not have the correct status when varying on the NWSD, the message CPDB1EF, "Status of network server host adapter NAME not valid" is reported. This is also the case if more than one of the local Host Adapters are specified in the NWSD. The host adapter failing is in the message.

The vary-on process for iSCSI connections is:

- ▶ Assume the associated NWSH device (Host Bus Adapter) is ACTIVE.
- ▶ A user starts the NWSD in one of the following ways:
 - Use “start” in iSeries Navigator.
 - Use the VRYCFG command.
 - Let it start by TCP/IP.
- ▶ IBM Director Server job (QCPMGTSVR) is started (if not already started).
- ▶ Call to IBM Director Server to check the state of the server.
- ▶ Allocate the NWSH resource to the NWSD.
- ▶ Write configuration files to the install drive (D:).
- ▶ Network storage spaces are allocated to the NWSD.
- ▶ Associated Ethernet lines are varied on.
- ▶ TCP/IP Interfaces are started.
- ▶ Call to IBM Director Server to power on the xSeries or Blade.
- ▶ The xSeries or Blade should come up.

6.8.1 Starting iSCSI integrated server during an iSeries IPL

You can use the TCP/IP interface created for Point-to-Point (PTP) Virtual LAN Line Description (LIND) attached to the NWSD to start your integrated Windows server each time that TCP/IP is started on the iSeries.

- ▶ **Set Autostart on a virtual PTP interface associated with the NWSD using iSeries Navigator:**
 - a. Open iSeries Navigator.
 - b. Expand **YourSystem** → **Network** → **TCP/IP Configuration** → **IPv4** and click **Interfaces**.
 - c. Right-click the virtual PTP interface associated with the NWSD in question defaulted to **YourNWSDNAMEPP** and select **Properties**.
 - d. Select the **Advanced** tab.
 - e. Check the box **Start interface when TCP/IP is started**. See Figure 6-71 on page 220.

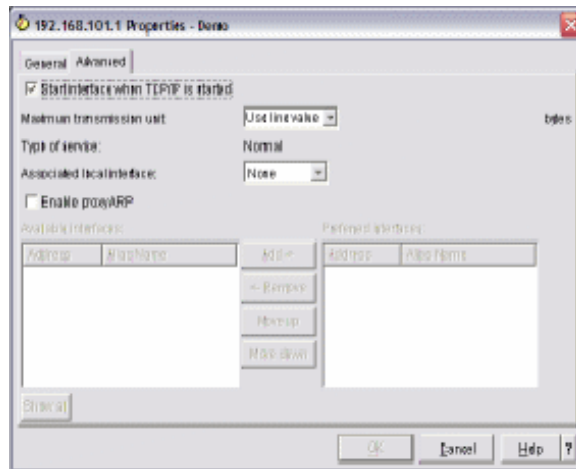


Figure 6-71 Properties TCP/IP interface

- **Set Autostart on a virtual PTP interface associated with the NWSD using CL command.**
 - a. On a 5250 command line, type `CFGTCP`, and press Enter.
 - b. The Configuration TCP/IP menu appears.
 - c. Type option 1 (Work with interfaces) on the command line.
 - d. The Work with interfaces display appears, scroll to the virtual PTP interface associated with the NWSD (line associated with the interface has the name **YourNWSDNAMEPP**) and put option 2 (Change) in front of it, and press Enter.
 - e. Change the parameter **Autostart** (AUTOSTART) to *YES, and press Enter.

Set the NWSH device to activate at IPL time

The NWSH device, which controls the Local Host Bus Adapter (HBA), should be active before the NWSD will be activated. The best way to do this to make sure it is activated during the IPL.

1. On a 5250 command line, type `WRKDEVD DEVD (*NWSH)`, and press Enter.
2. Scroll down to the NWSH you want to get online during IPL and put 2 (Change) in front of it and press Enter.
3. The Change Device Description (CHGDEVD) display appears. Change the parameter **Online at IPL** (ONLINE), and type *YES, and press Enter. On the next IPL, this device will be started.

Set TCP/IP to start when iSeries IPLs

In order to set TCP/IP to start when the iSeries IPLs:

1. On a 5250 command line, type `CHGIPLA` and press F4-prompt.
2. Change the Start TCP/IP parameter to *YES and press Enter.

6.8.2 Starting iSCSI integrated server within a CL Program

Starting an iSCSI integrated server in a CL program has its benefits, for instance, for backup procedures or shutting down the server before powering off the iSeries. You need to be aware that the Host Bus Adapter (HBA) devices associated with NWSD are varied on as well. So, when the NWSD has multiple NWSH objects configured, these need to be varied on.

```

0001.00 /*****
0002.00 */
0003.00 /* PROGRAM TO VARY ON MULTIPLE INTEGRATED XSERIES SERVERS */
0004.00 */
0005.00 /*****
0006.00     PGM
0007.00     MONMSG      MSGID(CPF0000)
0008.00     HLDJOBQ      JOBQ(QGPL/QBATCH)
0009.00     CHGJOBQE      SBSQ(QSYS/QBATCH) JOBQ(QGPL/QBATCH) MAXACT(16)
0010.00 /*****
0011.00 /* START ISCSI INTEGRATED SERVER */
0012.00 /*****
0013.00     VRYCFG CFGOBJ(AS300NWSH1) CFGTYPE(*DEV) STATUS(*ON)
0014.00     VRYCFG CFGOBJ(AS300NWSH2) CFGTYPE(*DEV) STATUS(*ON)
0015.00     DLYJOB DLY(60) /* WAIT 1 MINUTE */
0016.00     VRYCFG CFGOBJ(ANWSD) CFGTYPE(*NWS) STATUS(*ON)
0017.00     DLYJOB DLY(180) /* WAIT 3 MINUTES */
0018.00 /*****
0019.00 /* START OTHER ISCSI INTEGRATED SERVERS */
0020.00 /*****
0021.00     RLSJOBQ      JOBQ(QGPL/QBATCH)
0022.00     SBMJOB      CMD(VRYCFG CFGOBJ(BNWSH CNWSH) CFGTYPE(*DEV) +
0023.00                  STATUS(*ON)) JOBQ(QGPL/QBATCH)
0024.00     SBMJOB      CMD(VRYCFG CFGOBJ(BNWSH CNWSH) +
0025.00                  CFGTYPE(*NWS) STATUS(*ON)) JOBQ(QGPL/QBATCH)
0026.00 /*****
0027.00 /* START OTHER WINDOWS SERVERS (IXS OR IXA) */
0028.00 /*****
0029.00     SBMJOB      CMD(VRYCFG CFGOBJ(IXS2000A IXA2000B) +
0030.00                  CFGTYPE(*NWS) STATUS(*ON)) JOBQ(QGPL/QBATCH)
0031.00     ENDPGM
***** End of data *****

```

Figure 6-72 Start iSCSI integrated server, IXS servers, and IXA servers

6.8.3 Starting iSCSI integrated server from Service Processor

With an iSCSI integrated server, you are allowed to use **start** → **shutdown** from the Windows server console. At that time, the NWSD will not be varied off but stays on as *varied on*. The NWSD at status varied on is just waiting for an initiator to connect. So if the NWSD has the status *varied on*, you can restart the Windows server by just pressing the **power button** on the xSeries or the little **power button** at the top (under the cover) of the Blade within a BladeCenter. The other option is to start the server using the WEB interface of the service processor. There is a slight difference between a RSA II (xSeries) service processor and Management Module (BladeCenter) Web interface. It states when there is a difference between the two.

Note: There is not a way to start the NWSD from the xSeries or Blade to the **varied on** status. This still has to be done with the VRYCFG command or iSeries Navigator.

To start the iSCSI integrated server from the Web interface of the service processor (SP) of the xSeries or the Management Module (MM) of the BladeCenter, assuming the server is off, follow the steps (Internet explorer is used):

1. Set the IP address to something in the same subnet as the SP/MM IP address. The default SP/MM IP address is 192.168.70.125.
2. Open a Web browser. In the address or URL field, type the IP address (192.168.70.125) of the SP/MM to which you want to connect. The Connect To *<IP Address>* window will open.
3. Enter the user ID and password, which is known on the SP/MM, and press **OK**. The SP/MM default user ID is USERID and password is PASSWORD (where 0 is a zero).
4. Select a **time-out** value on the next window, and click **continue** or **Start new session** if there is another session active on another PC.
5. For Blades, follow these steps; otherwise, skip this step and continue with step 6.
 - a. Select at the left pane **Power/Restart** under **Blade Tasks**.
 - b. At the right pane where the Blades are shown, select the Blade to start.
 - c. At the end of the list, click **Power On Blade**.
 - d. A Microsoft Internet Explorer® window opens with the question, "Are you sure you want to proceed with this operation?" Click **Yes**.
 - e. Another Microsoft Internet Explorer window opens saying that it can take a few moments. Click **OK**.
 - f. The window refreshes and the Pwr column shows that the Blade is **ON**. See Figure 6-73 on page 223.
 - g. At this time, Windows boots and the NWSD becomes active. You can see the flow of the boot by doing remote control. This is described in 6.8.6, "Starting Remote Control on MM to see Windows booting" on page 229.

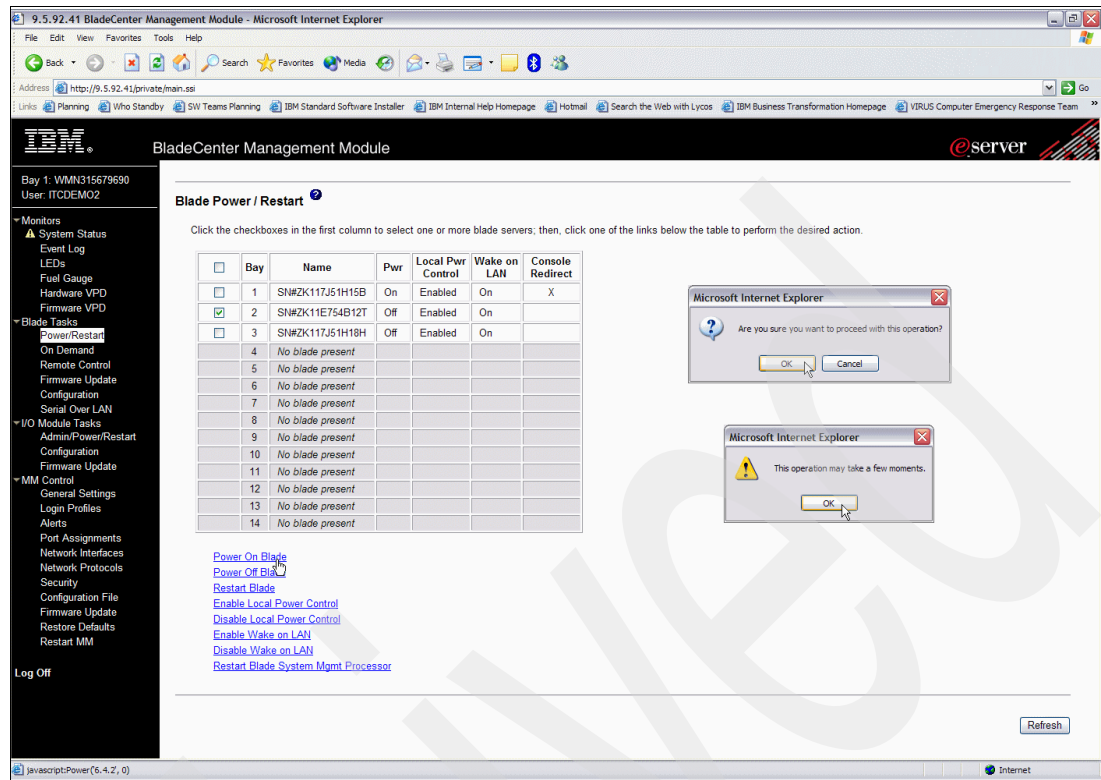


Figure 6-73 Power On Blade in Management Console

6. For xSeries (RSAII), follow these steps:
 - a. Select **Power/Restart** at the left pane under **Tasks**.
 - b. At the right pane, you will see the options and status of the xSeries.
 - c. Click **Power On Server immediately**.
 - d. A Microsoft Internet Explorer window opens asking you if you are sure you want to proceed with this operation. Click **Yes**.
 - e. The window refreshes and the Power status is **ON**. See Figure 6-74 on page 224.
 - f. At this time, Windows boots and the NWSD becomes active. You can see the flow of the boot by doing remote control. This is described in 6.8.7, “Starting Remote Control on RSA II to see Windows booting” on page 230.

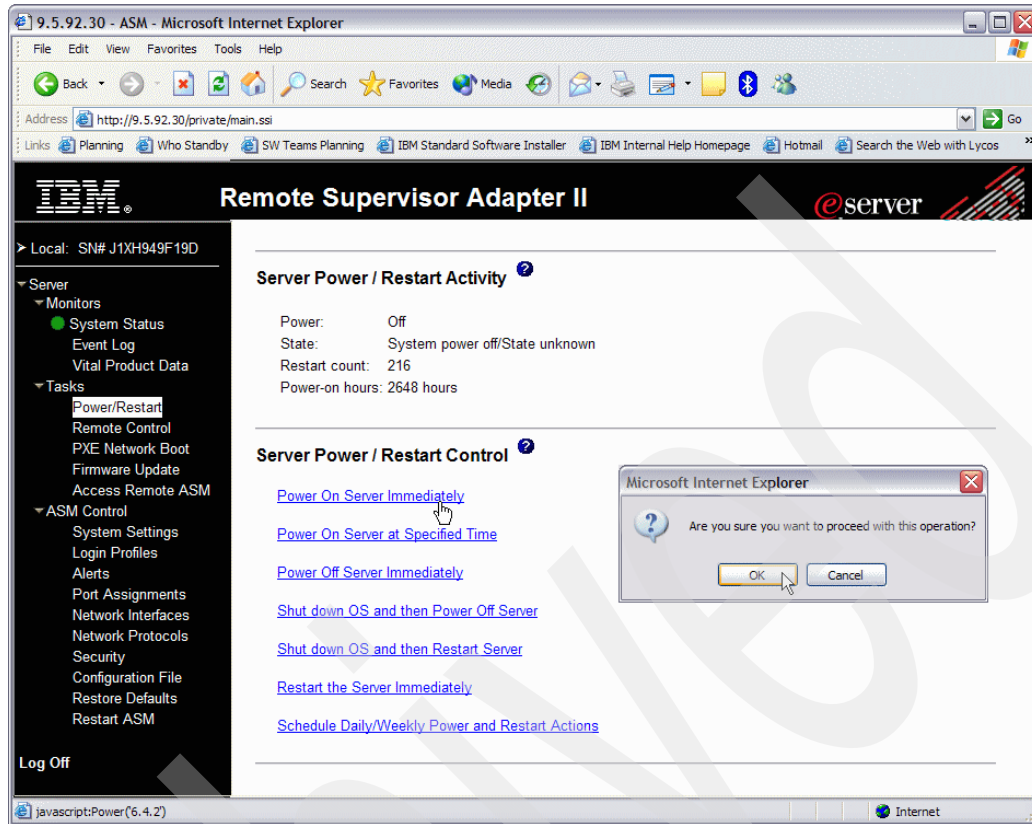


Figure 6-74 Power On xSeries in RSA II

7. For both RSA II and MM, you can select **Log Off** on the left pane at the left bottom side to close the session and close your Web browser window.

6.8.4 Starting iSCSI integrated server from iSeries Navigator

As we mentioned before, “Windows administration” under the topic “Network” has moved to a new topic called “Integrated Server Administration” right under “YourSystem” in iSeries Navigator Release 540. See Figure 6-75 on page 225.

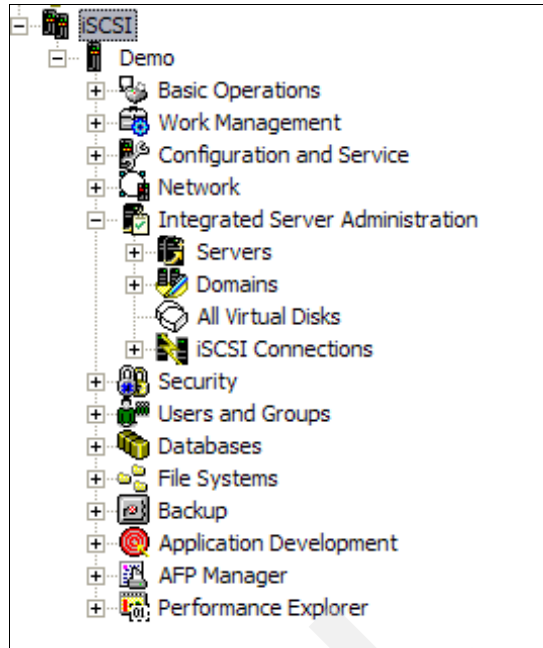


Figure 6-75 Integrated server administration in iSeries Navigator

There is a new section added for iSCSI, which is iSCSI connections. When this is expanded, you find the four objects, which are needed in an iSCSI environment. These are “Local Host Adapter” (NWSH), “Remote Systems” (RMTSYS), “Service Processors” (SRVPRC), and “Connection Security” (CNNSEC). The first one (NWSH) is a device, and the other three are part of the NWSCFG object, which also can be seen with the i5/OS command WRKNWSCFG. See Figure 6-76 on page 226.

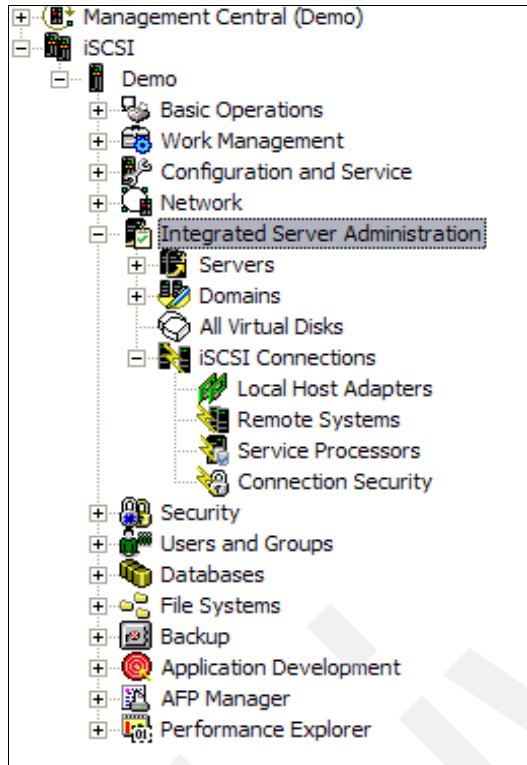


Figure 6-76 iSCSI Connections

To start all integrated servers, do the following:

1. Expand **YourSystem** → **Integrated Server Administration**. See Figure 6-77 on page 227.
2. Right-click **Servers** and select **Start All**. Another way is to select **Start all integrated servers** in the taskpad below. As soon as you click **Start All**, all are started, so no warning message appears.

Note: iSeries Navigator has several views such as Toolbar, Status Bar, and Taskpad. These can be shown or not shown by just clicking View and select or deselect the views.

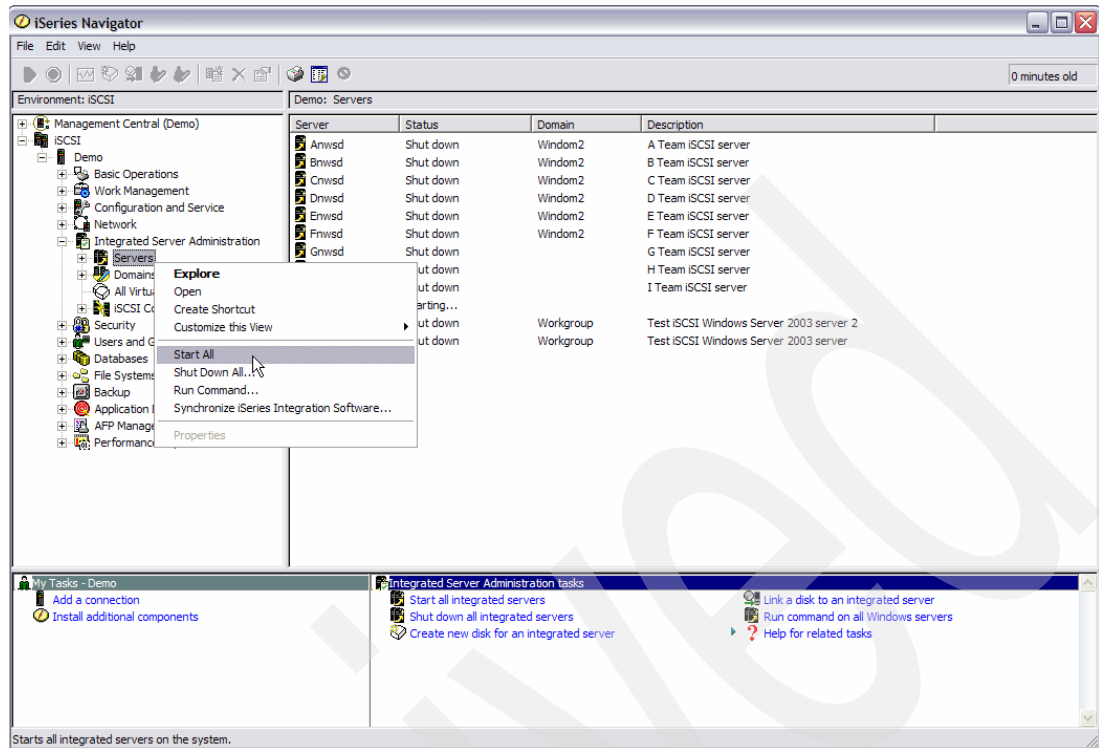


Figure 6-77 Start all integrated servers

To start one integrated server, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **Servers** or click **Servers**.
2. Right-click the integrated server to start (shutdown status), select **Start**. See Figure 6-78 on page 228.

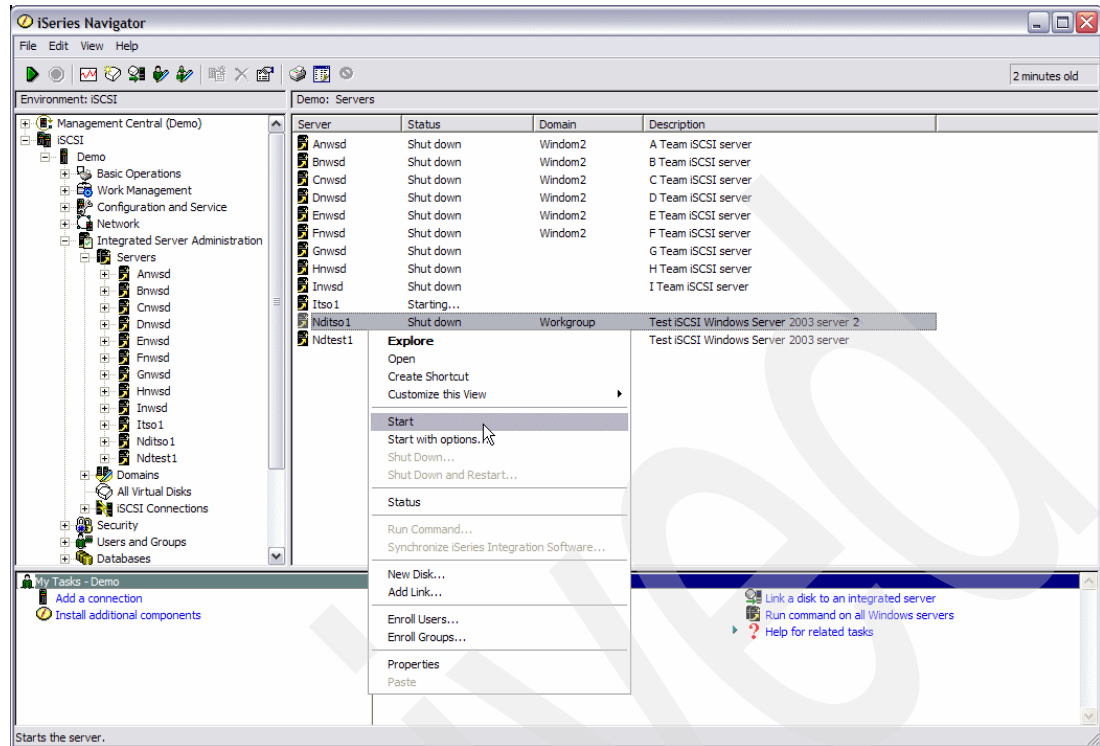


Figure 6-78 Start one integrated server

6.8.5 Starting iSCSI integrated server using CL command

Starting an iSCSI integrated server using CL command can be done with several commands.

Choose one of the following CL commands on a 5250 command line:

1. WRKNWSD

- On a command line, type WRKNWSD and press Enter.
- The Work with Network Server Descriptions display appears, scroll to the integrated server to start and enter option 8 (Work with Status).
- The Work with Configuration Status display appears, enter option 1 (Vary On) in the option column in front of the first entry, which is the NWSD, and press Enter. The integrated server starts.

2. WRKCFGSTS

- On a command line, type WRKCFGSTS *NWS and press Enter.
- The Work with Network Server Descriptions display appears, scroll to the integrated server to start and enter option 1 (Vary On) on the option column in front of the left most entry, which is the NWSD, and press Enter. The integrated server starts.

3. VRYCFG

- On a command line, type VRYCFG CFGOBJ (integrated server name) CFGTYPE(*NWS) STATUS(*ON), and press Enter. The integrated server starts (Figure 6-79 on page 229).

Vary Configuration (VRYCFG)		
Type choices, press Enter.		
Configuration object	ITS01	Name, generic*, *ANYNW...
+ for more values		
Type	*NWS	*NWS, *NWI, *LIN, *CTL...
Status	*ON	*ON, *OFF, *RESET...
Bottom		
F3=Exit	F4=Prompt	F5=Refresh
F24=More keys	F12=Cancel	F13=How to use this display

Figure 6-79 VRYCFG command to start an iSCSI integrated server

6.8.6 Starting Remote Control on MM to see Windows booting

The Management Module (MM) is the service processor for a BladeCenter. To start Remote Control on the Management Module (MM) of a BladeCenter to see the state of Windows or see it booting, follow the steps:

1. Set the IP address to something in the same subnet as the MM IP address. The default MM IP address is 192.168.70.125.
2. Open a Web browser. In the address or URL field, type the IP address (192.168.70.125) of the MM to which you want to connect. The **Connect To <IP address>** window opens.
3. Enter the user ID and password, which are known on the MM. Click **OK**. The MM default user ID is USERID and password is PASSWORD (where 0 is a zero).
4. Select a **time-out** value on the next window and click **continue** or **Start new session** if there is another session active on another PC.
5. The System Status window appears of the MM showing the status. Click **Remote Control** under **Blade Tasks**, the remote control window appears.
 - a. On this window, you can see the Remote Control status and read the functions, which can be disabled for Start Remote Control:
 - a. KVM owner represents which Blade has ownership of the keyboard, video, and mouse.
 - b. Media tray owner represents which Blade has ownership of the media such as CD drive and diskette drive.
 - c. Console Redirect informs you if there is already a remote control session running.
6. Click **Start Remote Control**, there are possibly two warning messages to run an application, click **Run** on these. The BladeCenter Remote Control window opens (Figure 6-80 on page 230), which is divided into three panes:
 - a. The Change KVM/Media Tray Owner pane is to manage which Blade has KVM owner, which is displayed in the Remote Console window, and which Blade has the media tray owner. By pressing on the left and right of the pull-down of KVM owner, you switch from one Blade to another. The Media tray owner should be selected with the pull-down menu.
 - b. The Remote Disk pane is to mount, for instance, your local CD-drive to the Remote Console session.
 - c. The Remote Console pane shows the window of the Blade selected at Change KVM/Media tray owner.

- To logoff, just close the Web browser with the Remote Control session with the **X** at the right top of the window and select **Logoff** on the left pane at the bottom to exit the session to the MM.

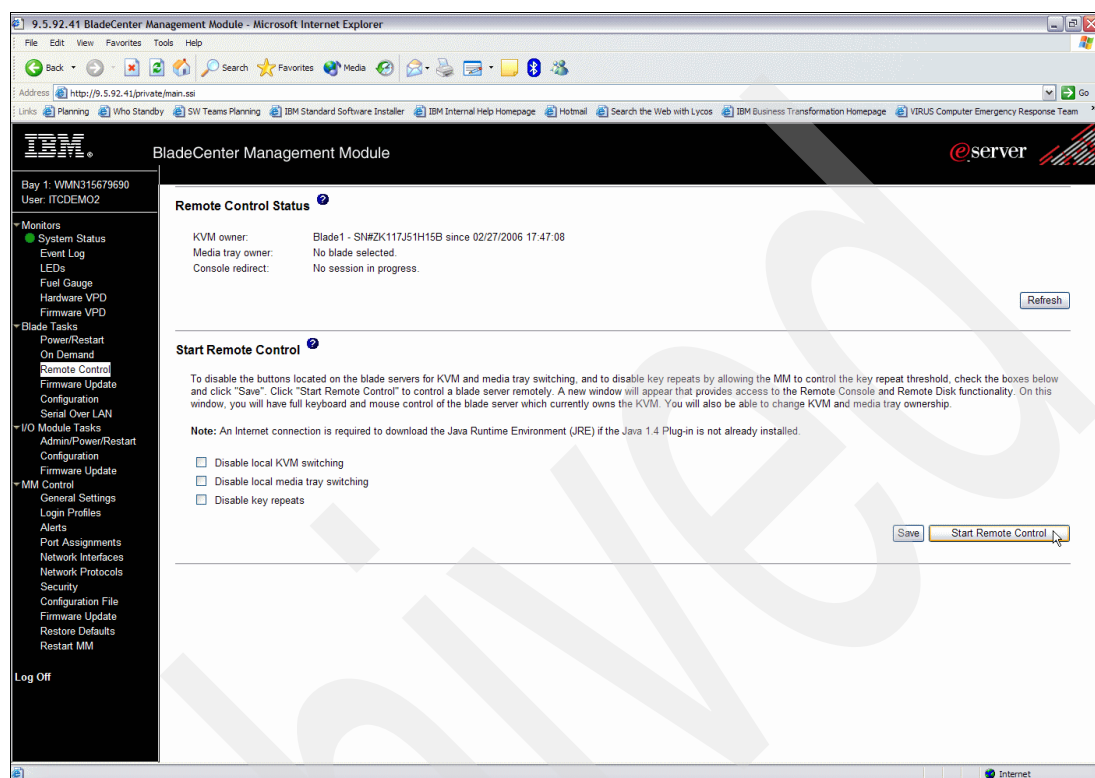


Figure 6-80 Start Remote Control in MM (BladeCenter)

6.8.7 Starting Remote Control on RSA II to see Windows booting

RSA II stands for Remote Supervisor Adapter II. This RSA is the service processor (SP) for an xSeries. To start Remote Control on the Remote Supervisor Adapter (RSA) of an xSeries to see the state of Windows or see it booting, follow the steps (Figure 6-81 on page 231):

- Set the IP address to something in the same subnet as the SP IP address. The default SP IP address is 192.168.70.125.
- Open a Web browser. In the address or URL field, type the IP address (192.168.70.125) of the SP to which you want to connect. The Connect To <IP Address> window opens.
- Enter the user ID and password, which are known on the SP, and press **OK**. The SP default user ID is USERID and password is PASSWORD (where 0 is a zero).
- Select a **time-out** value on the next window and click **continue** or **Start new session** if there is another session active on another PC.
- The System Status window appears of the SP showing the status of the xSeries. Click **Remote Control** under **Tasks**, the remote control window appears:
 - On this window, you can see the Remote Control status and the two options to start remote control, Start Remote Control in Single User Mode or Multi-User Mode. The difference between these two is that with Multi-User mode you are able to run multiple session and it provides remote disk access; however, Remote disk function does not support multi-users.

6. Click **Start Remote Control in Single User Mode** or **Start Remote Control in Multi-User Mode** to get the remote control session for yourself or for multiple users. There are possibly two warning messages to run an application, click **Run** on these. So then, you can mount your local PC CD-drive to the Remote console session. Only with Multi-User mode, the Remote Disk Function is not available for multiple users.
7. To log off, just close the Web browser with the Remote Control session with the **X** at the right top of the window, and select **Logoff** on the left pane at the bottom to exit the session to the SP.

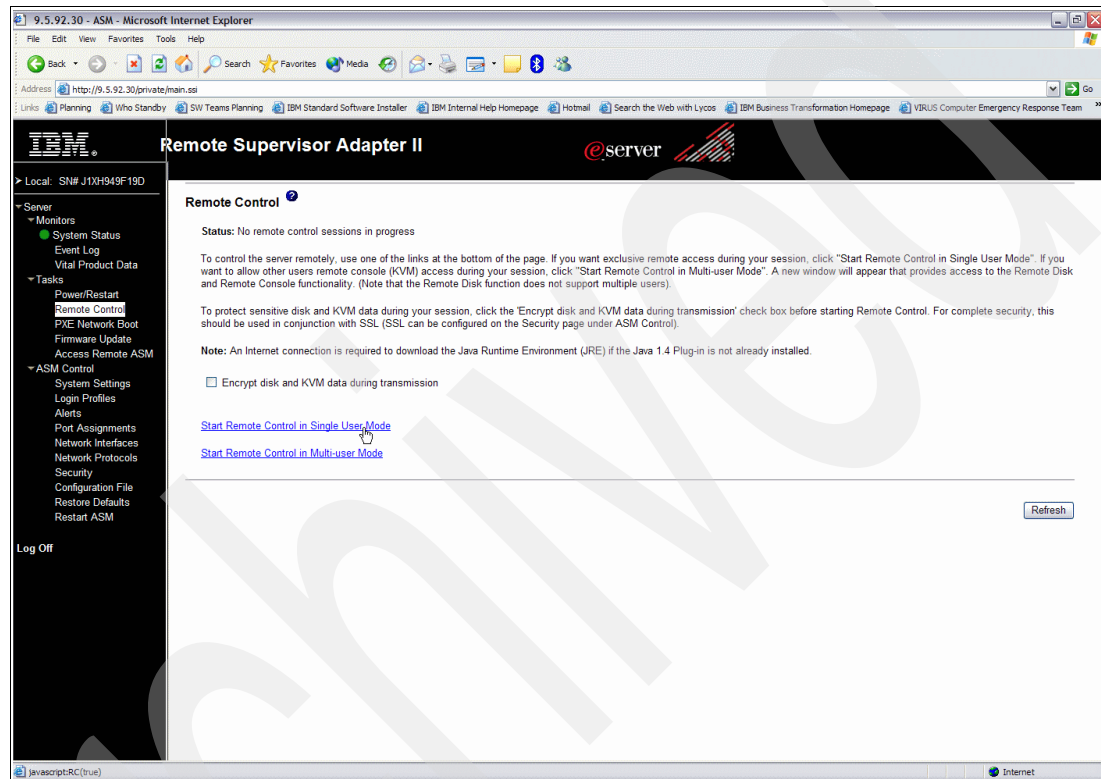


Figure 6-81 Start Remote Control on RSA II (xSeries)

6.9 Stopping iSCSI integrated server

Just as with starting the iSCSI integrated server, there are many ways to stop the iSCSI integrated server. If the Network Server Host Adapter (NWSH) device is varied off when there is an active iSCSI integrated server, the storage paths or Virtual Ethernet paths defined are unavailable at this time causing possible data loss. Within iSeries Navigator, you get a confirm window showing if there are any active NWSD using this NWSH. At this time, you can press **cancel** to not stop the NWSH. When **vrycfg** is used on a command line for an active NWSH with an active associated NWSD, you receive the message CPD5960, "Device NWSH-name cannot be varied off at this time."

Note: Remember that integrated Windows servers require that TCP/IP is active on the iSeries before varying on or off the server. You should always vary off your integrated Windows servers prior to running the ENDTCP command.

When executing a **vrycfg**, the vary off process initiates two concurrent ways to shut down the server:

- ▶ First a shutdown request is issued using the iSCSI connection.
- ▶ After 15 seconds, IBM Director Server issues a shutdown as well.

6.9.1 Stopping an iSCSI integrated server using Windows Desktop

Due to iSCSI concepts, it is possible to use **start** → **shutdown** from the Windows desktop without any problem. You can even start the iSCSI-connected server by just pressing the power button, if you start it this way. This is a major benefit for maintenance.

A restart is also possible. Use **start** → **shutdown** and select **restart** from the pull-down list Figure 6-82. Also, when you have configured Hibernate in the power options, this can be used from the pull-down window when doing **start** → **shutdown**, then the NWSD gets the status Varied on, which is normal in this case, but the virtual Point-to-Point Ethernet line and the TCP/IP interface stay active!



Figure 6-82 Shutdown from Windows Server 2003

When the shutdown from Windows desktop is done for an iSCSI integrated server, the NWSD on the iSeries does not vary off. The NWSD transitions from **active** to **varied on**.

Verify status using iSeries Navigator:

1. Expand **YourSystem** → **Integrated Server Administration**.
2. Click **Servers**, at the right the integrate servers are shown and their status, refresh can be done by pressing F5 or the refresh icon.

Verify status using CL command:

1. On a 5250 command line, type `WRKCFGSTS *NWS`.
2. Refresh the display with F5.

The Change NWS Configuration (CHGNWSCFG) display appears.

So, as long as the status of the NWSD has the status **varied on**, you can just press the power button on the xSeries or Blade within in the BladeCenter to start it again or use the service

processor of the xSeries or BladeCenter to start. The NWSD with status varied on means that it is waiting for a connection.

6.9.2 Stop/Restart iSCSI integrated server using iSeries Navigator

To stop all integrated servers, do the following:

1. Expand **YourSystem** → **Integrated Server Administration**.
2. Right-click **Servers** and select **Shut Down All**. See Figure 6-83. Another way is to select **Shut Down all integrated servers** in the taskpad below. As soon as you click **Shut Down All**, all are shutdown, so no warning message displays.

Note: iSeries Navigator has several views, such as the Toolbar, Status Bar, and Taskpad. These can be shown or not shown by just clicking View and selecting or deselecting the views.

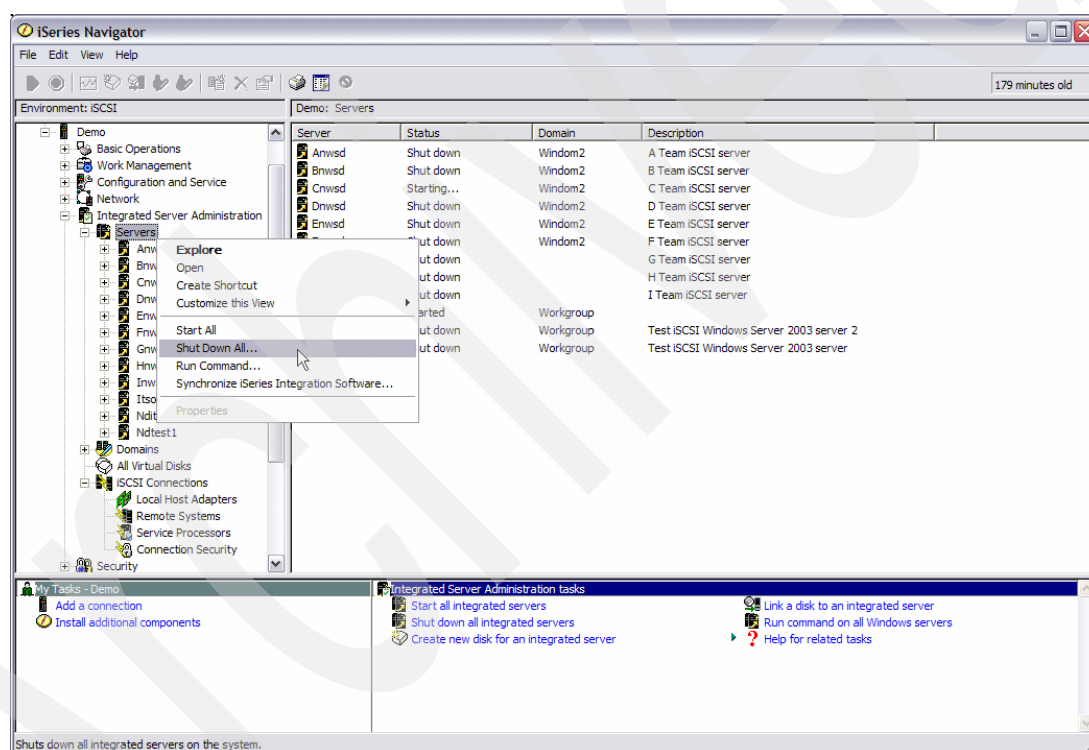


Figure 6-83 Shut Down All integrated servers

To stop one integrated server, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **Servers** or click **Servers**.
2. Right-click the integrated server to shut down (Active status), select **Shut Down**.
3. A Confirm Shut Down window opens (Figure 6-84 on page 234), you must click **Shutdown** to shut down the server.

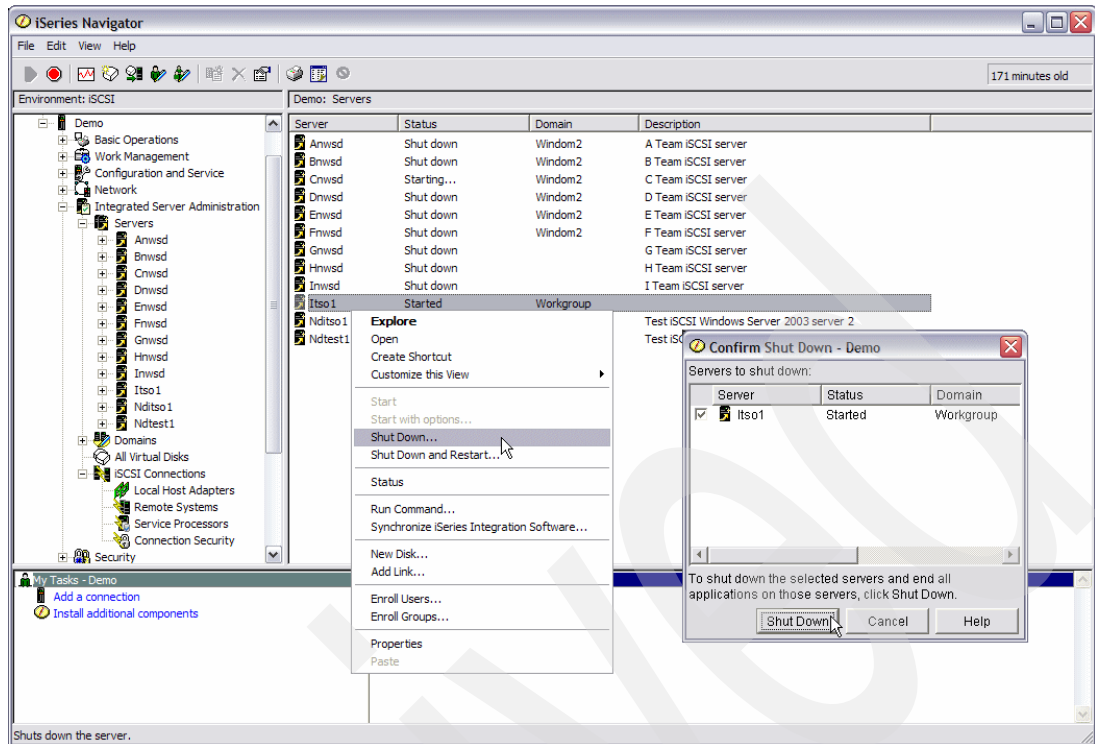


Figure 6-84 Shut down one integrated server

To restart an integrated server, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **Servers** or click **Servers**.
2. Right-click the integrated server to shutdown (Active status), select **Shut Down and Restart**.
3. A Confirm Shut Down and Restart window opens (Figure 6-85 on page 235), you must click **Restart** to restart the server.

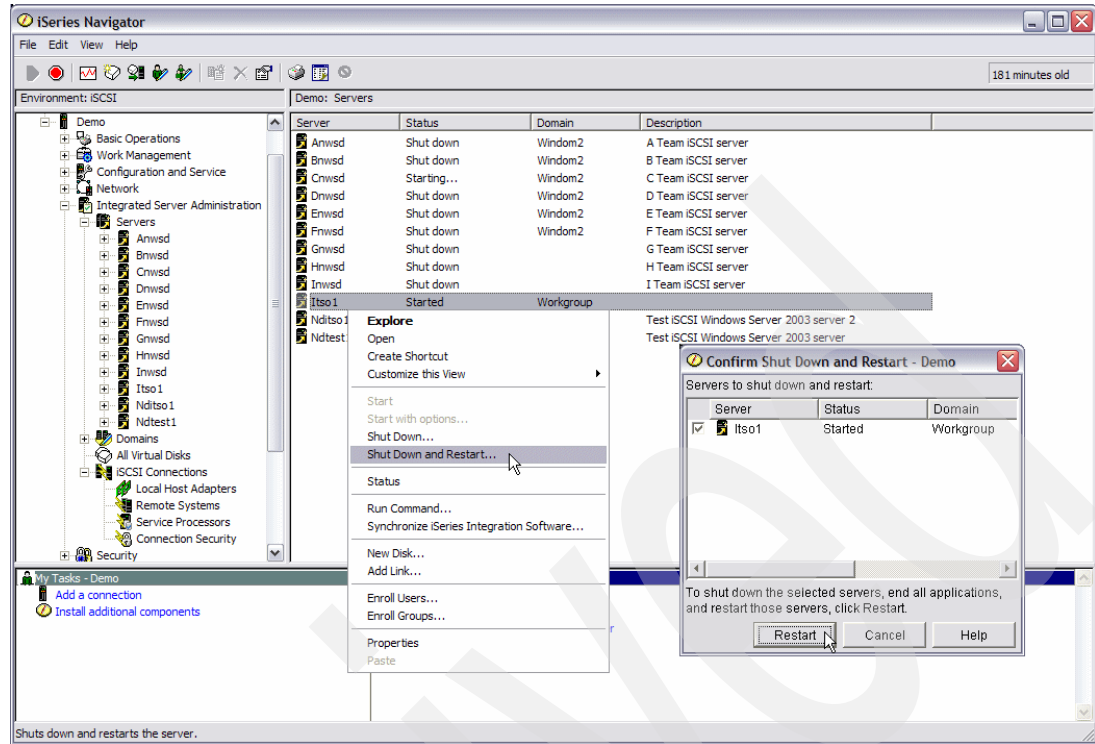


Figure 6-85 Shut Down and Restart integrated server

6.9.3 Stopping an iSCSI integrated server using CL commands

You can stop an iSCSI integrated server using CL commands with several commands.

Choose one of the following CL commands on a 5250 command line:

- ▶ **WRKNWSD**
 - a. On a command line, type WRKNWSD and press Enter.
 - b. The Work with Network Server Descriptions display appears, scroll to the integrated server to stop and enter option 8 (Work with Status).
 - c. The Work with Configuration Status display appears, enter option 2 (Vary Off) in the option column in front of the first entry, which is the NWSD, and press Enter. The integrated server stops.
- ▶ **WRKCFGSTS**
 - a. On a command line, type WRKCFGSTS *NWS and press Enter.
 - b. The Work with Network Server Descriptions display appears, scroll to the integrated server to stop and enter option 2 (Vary Off) on the option column in front of the left most entry, which is the NWSD, and press Enter. The integrated server stops.
- ▶ **VRYCFG**

On a command line, type VRYCFG CFGOBJ(*integrated server name*) CFGTYPE(*NWS) STATUS(*OFF) and press Enter. The integrated server stops (Figure 6-86 on page 236).

Vary Configuration (VRYCFG)		
Type choices, press Enter.		
Configuration object	ITS01	Name, generic*, *ANYNW...
+ for more values		
Type	*NWS	*NWS, *NWI, *LIN, *CTL...
Status	*OFF	*ON, *OFF, *RESET...
Bottom		
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display		
F24=More keys		

Figure 6-86 VRYCFG command to stop an iSCSI integrated server

6.9.4 Stopping integrated servers using a CL Program

Keep in mind the order in which you shut down your integrated servers. For instance, the Domain controller should be started first and stopped last. This ensures the Active directory integrity. The Network Server Host Adapter (NWSH) does not have to be set to varied off, because this is a device controller in the iSCSI HBA card.

```

0001.00 /*****
0002.00 /*
0003.00 /* PROGRAM TO VARY OFF MULTIPLE INTEGRATED XSERIES SERVERS
0004.00 /*
0005.00 /*****
0006.00     PGM
0007.00     MONMSG      MSGID(CPF0000)
0008.00     HLDJOBQ      JOBQ(QGPL/QBATCH)
0009.00     CHGJOBQE     SBSQ(QSYS/QBATCH) JOBQ(QGPL/QBATCH) MAXACT(16)
0010.00 /*****
0011.00 /* STOP OTHER ICSH INTEGRATED SERVERS
0012.00 /*****
0013.00     RLSJOBQ      JOBQ(QGPL/QBATCH)
0014.00     SBMJOB      CMD(VRYCFG CFGOBJ(BNWS CNWS) +
0015.00                  CFGTYPE(*NWS) STATUS(*OFF)) JOBQ(QGPL/QBATCH)
0016.00 /*****
0017.00 /* STOP OTHER WINDOWS SERVERS (IXS OR IXA)
0018.00 /*****
0019.00     SBMJOB      CMD(VRYCFG CFGOBJ(IXS2000A IXA2000B) +
0020.00                  CFGTYPE(*NWS) STATUS(*OFF)) JOBQ(QGPL/QBATCH)
0021.00     ENDPGM
***** End of data*****

```

Figure 6-87 Stop iSCSI integrated server and other IXS or IXA servers

6.9.5 Stopping an iSCSI integrated server using the MM Web Interface

Within the Management Module of a BladeCenter, the Power/Restart options are different as they are in the RSA II of an xSeries server. There is no option to first shut down the operating system and then power the Blade off. If you just click **Power Off Blade** after selecting one or more Blades, these get the power off status, but Windows Server 2003 is not shut down as it should be, so this is not an option to stop an iSCSI integrated server.

6.9.6 Stopping an iSCSI integrated server using the RSA II Web Interface

In the same way you are allowed to use **start** → **Shutdown** on a Windows Desktop to shutdown an iSCSI server, you are also allowed to use the Web interface to the service processor of the xSeries, which is RSA II. To shut down an iSCSI server, follow the steps:

1. Follow the first four steps described in 6.8.7, “Starting Remote Control on RSA II to see Windows booting” on page 230.
2. The System Status window appears of the SP. Click **Power/Restart** under **Tasks**.
3. The **Power/Restart** window appears. On this window, you can select several Power functions. Select **Shut Down OS and then Power Off Server** to shut down the operating system first and then power off the xSeries. First a question asks, “Are you sure you want to proceed with this operation”? See Figure 6-88. Select **OK**.
4. There is also an option to restart, which is Shut Down OS and then Restart Server.

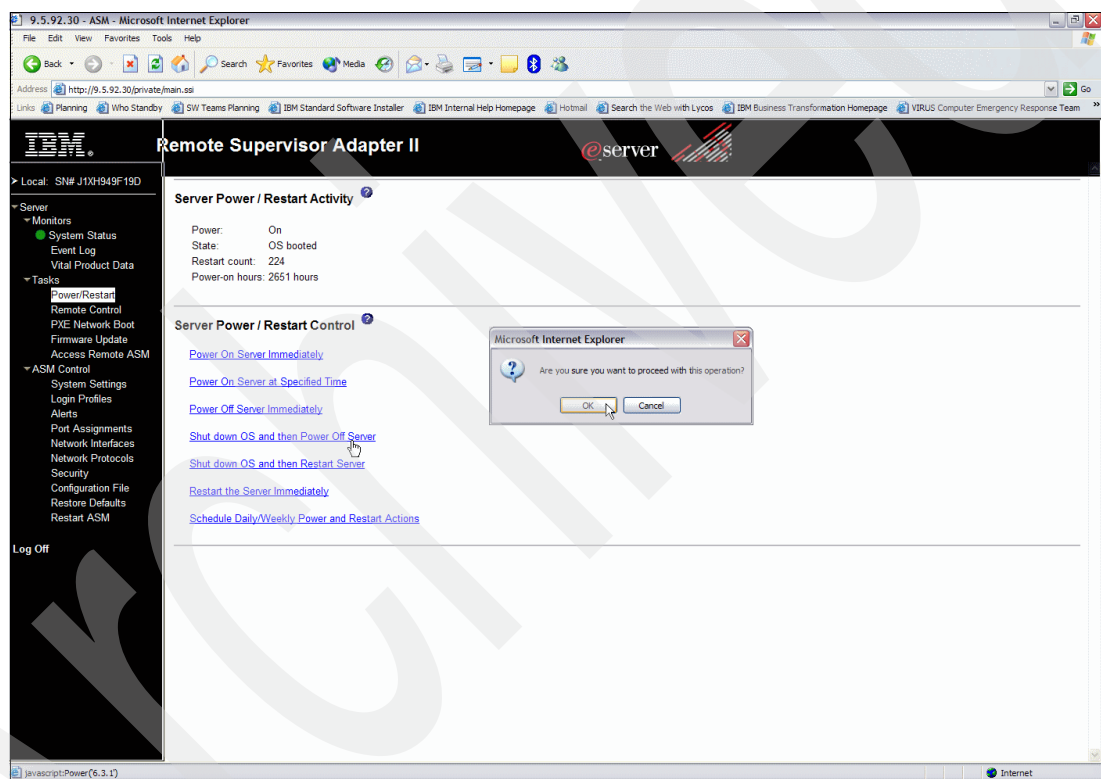


Figure 6-88 Power/Restart on RSAII

6.10 Status iSCSI integrated server

Status of an iSCSI integrated server can be seen through different interfaces and commands. You can display the status of the Network Server Description (NWS), but you can also request the status of the xSeries or Blade from iSeries Navigator or CL command. IBM Director Server is used for retrieving the status of the xSeries or Blade. Another way is to use the Web interface pointing to the SP or MM of the xSeries or BladeCenter.

6.10.1 Status of iSCSI integrated servers using iSeries Navigator

The status reflects the status as seen with the CL command WRKNWSSTS and not the status of the NWSD. To check the status of an iSCSI integrated server NWSD using iSeries Navigator, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration**.
2. Click **Servers** at the right pane and all integrated servers display. By default, the second column has the status of the servers. The status could be, for instance, Started, Starting, or Shutdown. These columns can be modified by right-clicking at the right pane and selecting **Customize this view** and selecting which columns and in which order they are presented.

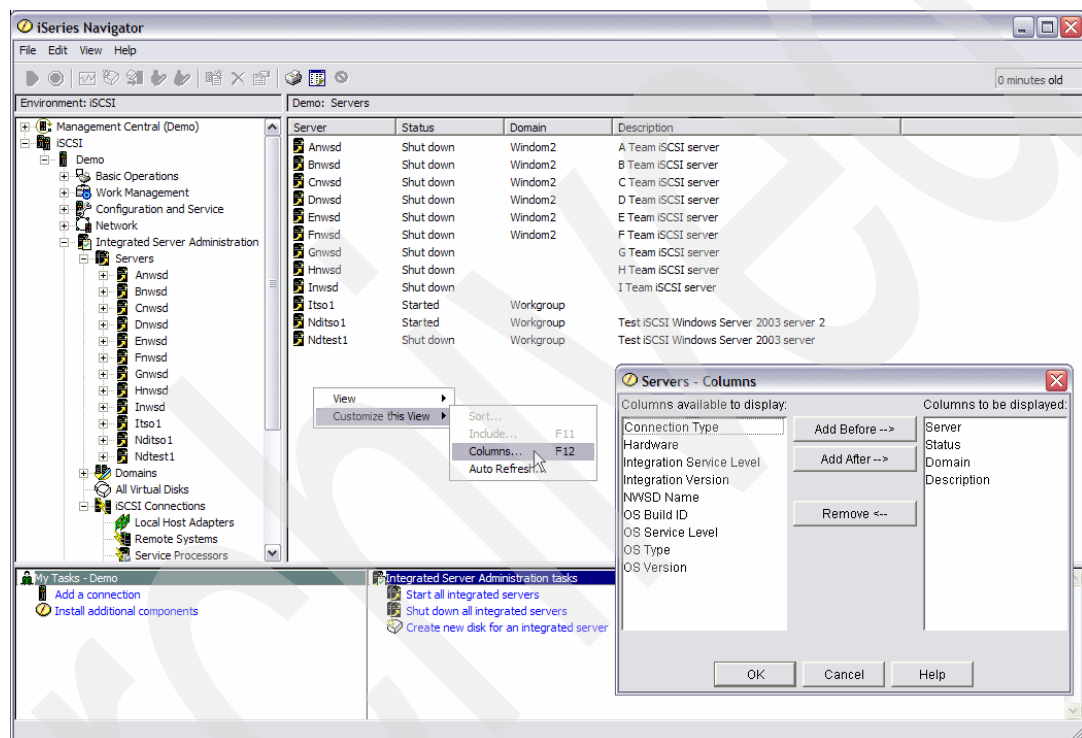


Figure 6-89 Status All integrated servers and customize column in iSeries Navigator

You can also display the status of one of the integrated servers to see, for example, the amount of connected users and processor utilization. To check these, follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **Servers**.
2. The integrated servers are displayed under **Servers**. Right-click a server and select **Status**. If it is active, it shows the connected users and so on. If not, it just shows you the status Shut down and a button to go to the System Operator message queue. See Figure 6-90 on page 239.

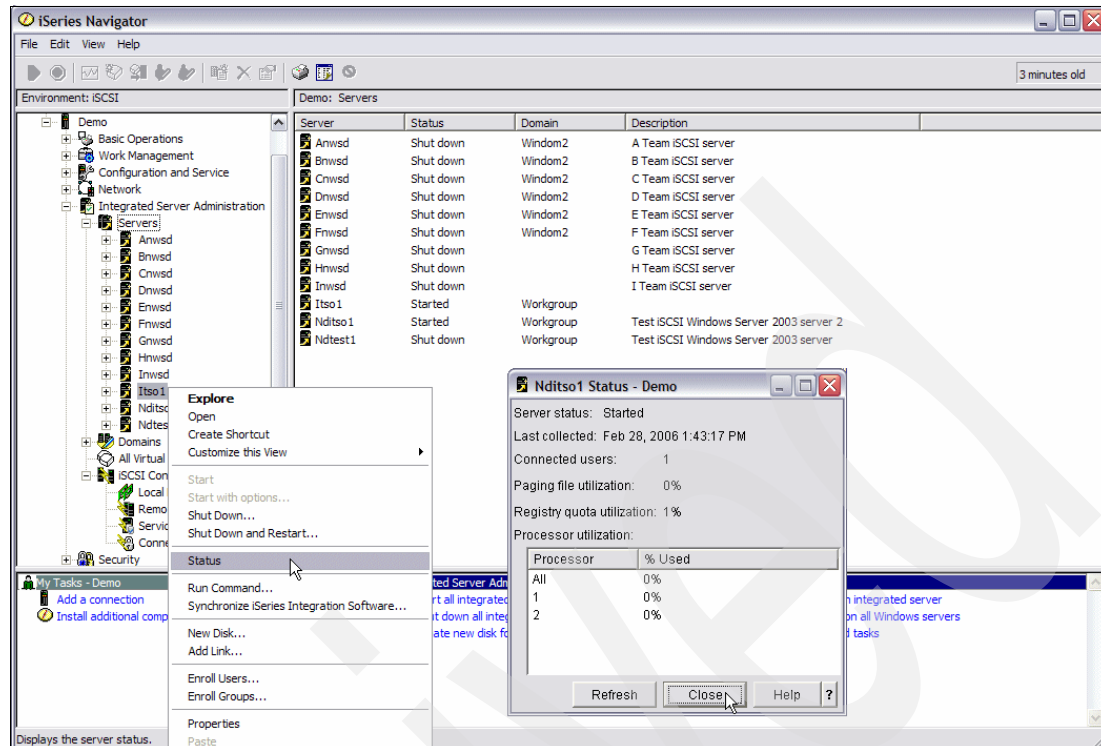


Figure 6-90 Detailed Status of integrated server in iSeries Navigator

6.10.2 Status of iSCSI integrated servers using CL commands

To check the status using CL commands, which reflect the status seen in iSeries Navigator, you have to use the command WRKNWSSTS. For the NWSD status, you need another command such as WRKCFGSTS *NWS.

To check the status of the NWSD, follow the steps:

1. On a command line, type WRKCFGSTS *NWS, and press Enter.
2. The Work with Configuration Status display appears. This shows the status the NWSD is in, such as Varied Off, Vary On, or Active. You can scroll to the integrated server you want to check the status for.

To check the status of the server, follow the steps (reflects the status as in iSeries Navigator):

1. On a command line, type WRKNWSSTS and press Enter or F4-Prompt. When you prompt, two parameters can be filled in:
 - a. Server: The Server name you want to display.
 - b. Server Type: The Server type. In this case, the option should be *Windows.
2. The Work with Network Server Status display appears (Figure 6-91 on page 240 and Figure 6-92 on page 240). This shows the status of the servers, such as inactive, Active, or pending. You can scroll to the integrated server you want to check the status for. A nice option on this window is option 5-Display Details (only valid when the server is active). It shows useful information such as:
 - a. Windows Server Status: Connected users, Paging file utilization, and Registry quota utilization
 - b. Windows Server Version: Version, Build Number, and Service Pack level

- c. AS400 Integration with Windows server version: Version, Language version, and Service pack Level (which is installed on the Windows side)
- d. Windows server processor utilization

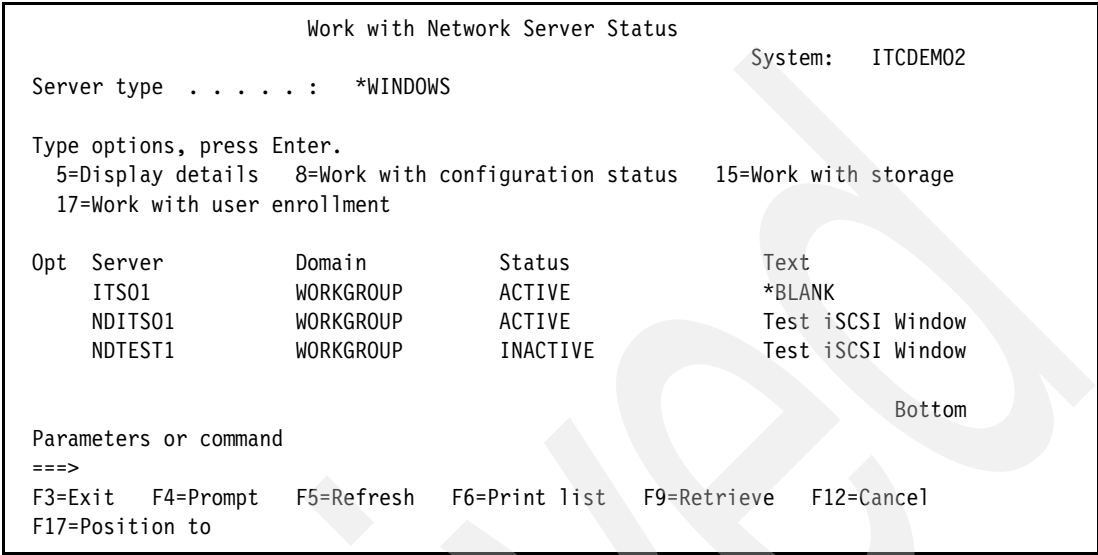


Figure 6-91 WRKNWSSTS display

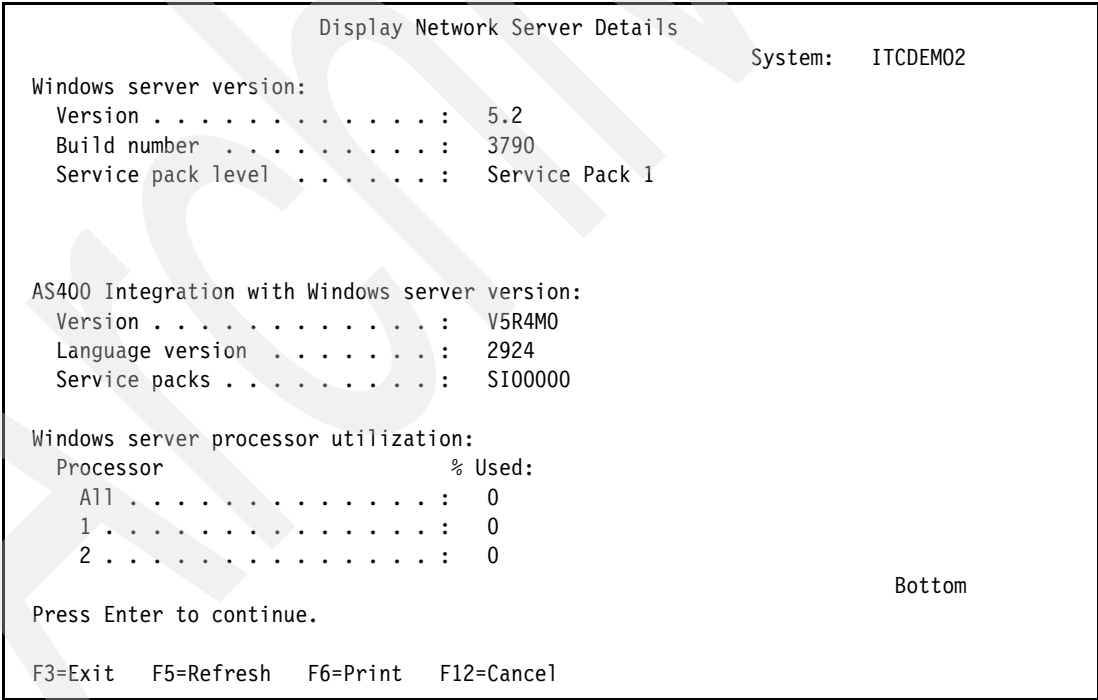


Figure 6-92 WRKNWSSTS option 5

6.10.3 xSeries or BladeCenter status using iSeries Navigator/CL commands

You can display status of an xSeries or Blade by retrieving the status of this server using the Remote System configuration (RMTSYS) using iSeries Navigator as well with the CL command.

To display the status of an xSeries or Blade using iSeries Navigator, go to 6.6.1, “Manage RMTSYS object using iSeries Navigator” on page 185.

To display the status of an xSeries or Blade using CL command, go to 6.6.2, “Manage RMTSYS object using CL command” on page 195.

6.10.4 Status of Blade using Web Interface

The status of the Blade can also be displayed using a Web browser pointing to the management module (MM) of the BladeCenter Chassis. This is described up to step 5 in 6.8.6, “Starting Remote Control on MM to see Windows booting” on page 229.

6.10.5 Status of xSeries using Web Interface

The status of the xSeries can also be displayed using a Web browser pointing to the service processor (RSA II) of the xSeries. You can also see the state the server is, for example, OS booted, meaning Windows Server 2003 is up and running. This is described up to point 5 in 6.8.7, “Starting Remote Control on RSA II to see Windows booting” on page 230.

6.10.6 Status of the NWSD degraded

There is a new status added for the NWSD, which is *degraded*. If the NWSD has the status degraded, there is a problem with one or more storage paths this NWSD is using. So there could be a problem with one of the Network Server Host Adapters (NWSH). Check if these are varied on. For more information, see the following Web site:

<http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzahq/rzahqproblemswithiscsiattached.htm>

Scroll down on the above Web page until you see The NWSD status is DEGRADED.

6.11 Manage Storage Spaces with iSCSI integrated server

With i5/OS R540, there are some new functions and limitations concerning storage spaces. The new function is to expand a storage space without copying the network storage space for non-system drives. For the system drive, there is still a copy. The necessary steps are described later in this section. This new function can also be used for an IXS or IXA server. The new limitation is that up to 64 network server storage spaces can be attached to an iSCSI integrated server and each of these network server storage spaces can have a maximum capacity of 1000 GB (1 TB). So the maximum you could have is slightly more than 60 TBs of storage space attached to a server. With iSCSI, the only link type possible is Dynamic, meaning you can add disk drive while the server is up or down.

The storage spaces you create goes in the integrated file system (IFS) directory /QFPNWSSTG.

6.11.1 Expanding a system drive (C:) using iSeries Navigator

Important: Make sure you have a recent backup of the C: drive before expanding.

To begin:

1. Shut Down the server as described in 6.9.2, “Stop/Restart iSCSI integrated server using iSeries Navigator” on page 233.
2. Expand **YourSystem** → **Integrated Server Administration** → **Servers** → **YourServer** or click **YourServer**.
3. Click **Linked Virtual Disks**, at the right pane, the list of linked virtual disks appears.
4. Right-click **the system drive (YourServer1)**, which is C: and select **New Based On**.
5. The New Disk Based On window opens (Figure 6-93). Type and select the following and select **OK**:
 - a. Enter a new Disk drive name and a new Description with words similar to New system drive.
 - b. Leave the Initialize disk with data from another disk, Source disk, Capacity, Disk Pool, and Planned file system to their default values.
 - c. Select **Link disk to server** and select the following using the pull-down menu:
 - i. **Link to server:** same server as the system drive is linked to
 - i. **Link type:** Dynamic (default)
 - ii. **Link sequence position:** Next available number (default)
 - iii. **Storage path:** Default 1 (NWSH name)
 - iv. **Access to disk drive:** Exclusive - Update (default)

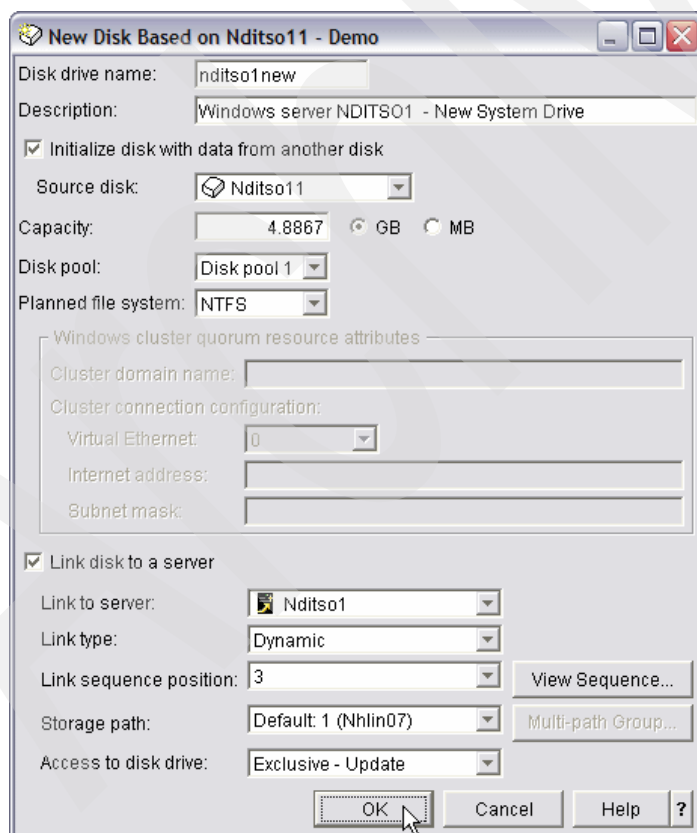


Figure 6-93 New Disk Based details window

6. A New Disk Based On status window opens, explaining that it is creating the disk. This can take some time depending on the capacity and your system.

7. When step 6 is finished, right-click the **new created** disk drive at the right pane and select **Properties**.
8. The Properties window opens of the new created disk, click the **Capacity** tab (Figure 6-94), and enter the **New capacity** at the bottom.

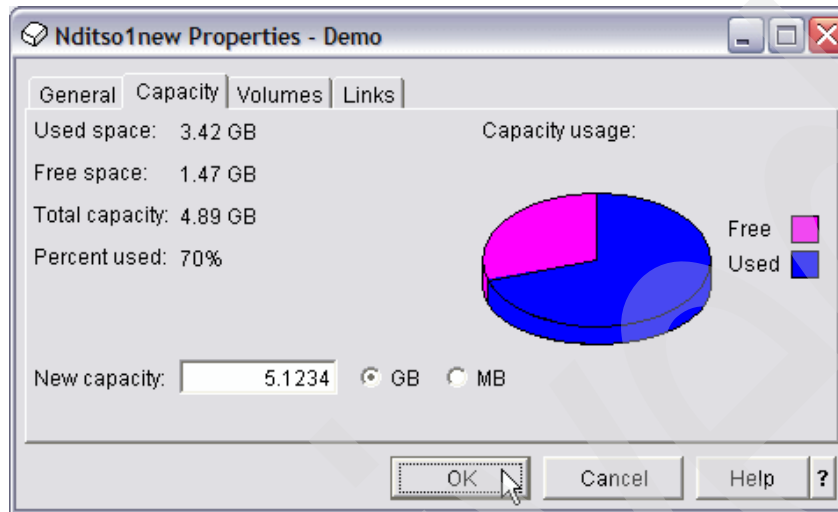


Figure 6-94 Capacity tab properties new created disk

9. A Confirm Change Disk Drive window opens (Figure 6-95): it reads that the drive is linked and it will be temporarily not be available, because it unlinks, expands with the capacity, and links it again. Select **Change**.

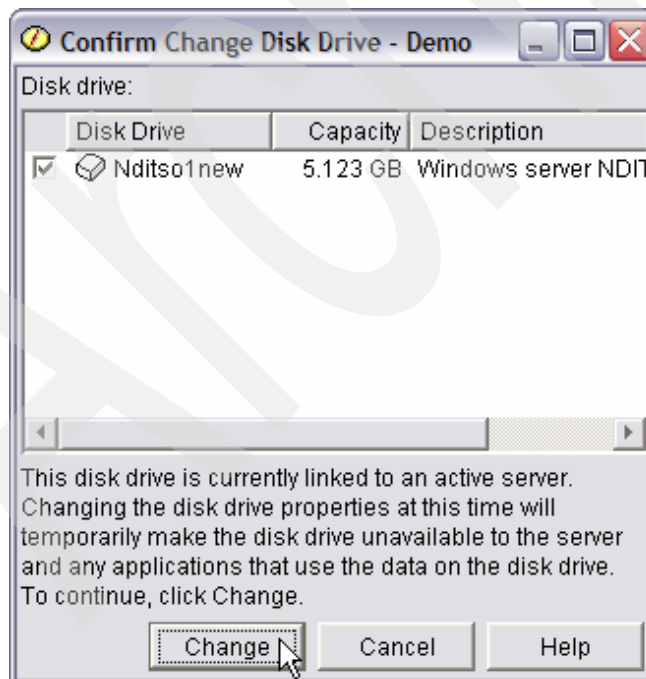


Figure 6-95 Confirm Change Disk Drive window

10. Another status window opens stating it is expanding the disk. When this is done, another window opens stating the drive is expanded to the capacity specified and disk

management must be done on Windows to make the added disk space is available to the server, click **OK**.

11. Start the integrated server as described in 6.8.4, “Starting iSCSI integrated server from iSeries Navigator” on page 224. When it is up and running, use, for example, the Remote Control function described in 6.8.7, “Starting Remote Control on RSA II to see Windows booting” on page 230, or 6.8.6, “Starting Remote Control on MM to see Windows booting” on page 229 to do the Diskpart utility on Windows Server 2003.
12. On Windows Server 2003, the diskpart utility should be done to get the expanded disk space added to the current disk space of the disk drive. In Windows Server 2003 disk management, you notice that the added disk space is not yet part of the current disk space. Follow the steps to perform Diskpart:
 - a. Open a DOS prompt by selecting a shortcut or **Start** → **Run** and enter **CMD**.
 - b. On the command prompt, type **Diskpart** and press Enter. The prompt changes from C:\....> to DISKPART>, you are now in the Diskpart utility.
 - c. Type **list volume**, a list with volumes appears. You see all the drives and their drive letters.
 - d. Select the new created volume by typing **select volume x**, where **x** is the volume number. A message returns saying the volume is selected.
 - e. Type **extend** to expand the drive with the added disk space. It returns with the message “DiskPart successfully extended the volume”.

```
F:\Documents and Settings\Administrator>diskpart
Microsoft DiskPart version 5.2.3790.1830
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NDITS01

DISKPART> list volume

Volume ###  Ltr  Label          Fs      Type          Size      Status       Info
-----
Volume 0      E             DUD-ROM         0 B        Healthy
Volume 1      F             DUD-ROM         0 B        Healthy
Volume 2      C      NDITS011       NTFS      Partition     4997 MB     Healthy      Boot
Volume 3      D      NDITS012       FAT       Partition     805 MB     Healthy
Volume 4      G      NDITS011       NTFS      Partition     4997 MB     Healthy

DISKPART> select volume 4
Volume 4 is the selected volume.

DISKPART> extend
DiskPart successfully extended the volume.
```

Figure 6-96 Diskpart in DOS Window

13. Now that the new system drive is extended, we need to unlink the “old” system drive and link the “new” one. For this, because this concerns the system drive, the server should be shut down. Shutting down the server is described in 6.9.2, “Stop/Restart iSCSI integrated server using iSeries Navigator” on page 233.
14. When the server is Shut Down, do the following to unlink the “old” system drive and link the “new” expanded disk drive:
 - a. Expand **YourSystem** → **Integrated Server Administration** and click **All Virtual Disks**.
 - b. Right-click the “old” system drive and select **Remove Link**, a Remove Link from Server window opens, click **Remove**. The same procedure needs to be done for the “New” expanded system drive.

- c. Right-click the **“New” expanded** system drive and select **Add Link**, the Add Link to Server window opens (Figure 6-97). Using the pull-down list, select the following and click **OK**:
 - i. **Link to server:** same server as the “old” system drive was linked to
 - i. **Link type:** Dynamic (default)
 - ii. **Link sequence position:** 1 because it is the system drive
 - iii. **Storage path:** Default 1 (NWSH name)
 - iv. **Access to disk drive:** Exclusive - Update (default)

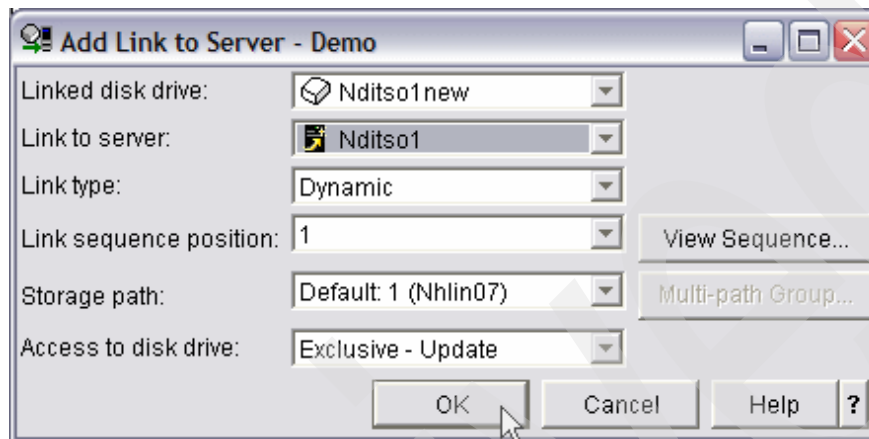


Figure 6-97 Add link new system drive

15. Start the integrated server as described in 6.8.4, “Starting iSCSI integrated server from iSeries Navigator” on page 224. The server starts with the new expanded system drive.

6.11.2 Expanding a system drive (C:) having multiple integrated servers

To expand a system drive (C:) that has more than one integrated server is almost the same procedure as having only one integrated server. The difference is that you do not have to make a copy of the “old” system drive. The order to do this is as follows, for details about how to do the steps, go to 6.11.1, “Expanding a system drive (C:) using iSeries Navigator” on page 241:

1. Shut Down the server for which you want to expand the system drive (C:) described in 6.9.2, “Stop/Restart iSCSI integrated server using iSeries Navigator” on page 233.
2. Unlink the “old” system drive (C:) described in “Expanding a system drive (C:) using iSeries Navigator” on page 241, steps 14a and 14b.
3. Expand the “old” system drive (C:) with the new capacity as described in “Expanding a system drive (C:) using iSeries Navigator” on page 241, step 8.
4. Link the expanded system drive (C:) to another integrated server which might be active, as follows:
 - a. Right-click the **“New” expanded system drive** and select **Add Link**, the Add Link to Server window opens. Using the pull-down list, select the following and click **OK**:
 - i. **Link to server:** the other (temporary) server
 - i. **Link type:** Dynamic (default)
 - ii. **Link sequence position:** next available for that server, leave the default
 - iii. **Storage path:** Default 1 (NWSH name)
 - iv. **Access to disk drive:** Exclusive - Update (default)
5. On this other temporary server, perform the Diskpart utility described in “Expanding a system drive (C:) using iSeries Navigator” on page 241, step 12.

6. Unlink the “new” expanded system drive from the other temporary server, same as step 2.
7. Link the “new” expanded system drive to the original server described in “Expanding a system drive (C:) using iSeries Navigator” on page 241, step 14c.
8. Start the integrated server with the new expanded disk described in 6.8.4, “Starting iSCSI integrated server from iSeries Navigator” on page 224.

6.11.3 Expand a Disk (non-system) using iSeries Navigator

To expand a disk (non-system) using iSeries Navigator:

1. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks**.
2. Right-click a drive, which you want to expand, and select **Properties**.
3. The Properties window opens, select the Capacity tab.
4. Enter the increased disk size in the **New capacity** field, and click **OK**.
5. The Confirm Change Disk Drive window opens. It reads that the drive is linked and it will be temporarily unavailable, because it unlinks it, expands with the capacity, and links it again. Click **Change**. A status opens saying that it is expanding the drive.
6. When it finishes, the drive was unlinked, expanded, and linked automatically. An information window opens that the drive is expanded to the capacity specified and disk management must be done on Windows to make the added disk space available to the server.
7. On Windows Server 2003, the diskpart utility should be done to get the expanded disk space added to the current disk space of the disk drive. In Windows Server 2003 disk management, you notice that the added disk space is not yet part of the current disk space. Follow the steps to perform Diskpart:
 - a. Open a DOS prompt by selecting a shortcut or **Start** → **Run**, and enter CMD.
 - b. On the command prompt, type Diskpart and press Enter, the prompt changes from C:\...> to DISKPART>. You are now in the Diskpart utility.
 - c. Type list volume, a list with volumes appears. You see all the drives and their drive letters.
 - d. Select the new created volume by typing select volume x, where x is the volume number. A message returns saying that the volume is selected.
 - e. Type extend to expand the drive with the added disk space. It returns with the message “DiskPart successfully extended the volume”.
8. The Disk drive can now be used with the expanded disk space.

6.11.4 Expanding system drive (C:) using CL command

With the CL command, the drive is not automatically unlinked, expanded, and linked again as it is with iSeries navigator:

1. Shut Down the server as described in 6.9.3, “Stopping an iSCSI integrated server using CL commands” on page 235.
2. On a command line, type WRKNWSSTG and press Enter, scroll to the system drive (C:), which you want to expand, and put option 3 (Copy) in the option column in front of it and press Enter.
3. The Create NWS Storage Space (CRTNWSSTG) display appears (Figure 6-98 on page 247). Type a new name for **Network server** storage space and type a description for **Text 'description'**, leave the Size as it was, it will expand later on, and press Enter. A

message appears saying that it creates the storage space. This can take some time depending on the capacity and the system.

Create NWS Storage Space (CRTNWSSTG)

Type choices, press Enter.

Network server storage space . .	nditsoln1	Name
Size	> 5246	*CALC, 1-1024000 megabytes
From storage space	> NDITS01	Name, *NONE
Auxiliary storage pool ID . . .	> 1	1-255
ASP device		Name
Text 'description'	New Expanded System Drive	

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 6-98 CRTNWSSTG display

4. Type WRKNWSSTG if not already on this display. Scroll to the new created system disk and put option 2 (Change) on the option column in front of it and press Enter. The Change NWS Storage Space (CHGNWSSTG) display appears (Figure 6-99). For **Size** (NWSSIZE), type the new capacity in MB. And press Enter.

Change NWS Storage Space (CHGNWSSTG)

Type choices, press Enter.

Network server storage space . .	> NDITS01N1	Name
Size	5300	*CALC, 1-1024000 megabytes
Text 'description'	*SAME	

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 6-99 CHGNWSSTG display

5. A status message appears that it creates the storage space. So, the system disk is now expanded. Link the “new” system drive to the same iSCSI server with a sequence number other than 1 or 2, this is done automatically cause the “old” system drive is still linked. Type option 10 (Add link) on the option column in front of the “new” system drive on the Work with Network Server Storage Spaces display by running the command WRKNWSSTG.
6. The Add Server Storage Link (ADDNWSSTGL) display appears (Figure 6-100 on page 248). Type for **Network Server Description** (NWSD) the same as where the “old” system drive is linked and for **Dynamic storage link** (Dynamic), type *YES. Three other parameters appear. Leave these as the defaults, and press Enter.

Add Server Storage Link (ADDNWSSTGL)

Type choices, press Enter.

Network server storage space . . . NWSSTG	> NDITS01N1
Network server description . . . NWSD	> NDITS01
Dynamic storage link DYNAMIC	> *YES
Access ACCESS	*UPDATE
Drive sequence number DRVSEQNBR	*CALC
Storage path number STGPTHNBR	*DFTSTGPTH

Bottom

F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display
F24=More keys

Figure 6-100 Add Server Storage Link window

- A status message shows saying “Network server storage space link added”. Start the integrated server described in 6.8.5, “Starting iSCSI integrated server using CL command” on page 228. When it is up and running, use, for example, the Remote Control function described in 6.8.7, “Starting Remote Control on RSA II to see Windows booting” on page 230 or 6.8.6, “Starting Remote Control on MM to see Windows booting” on page 229 to do the Diskpart utility on Windows Server 2003.
- On Windows Server 2003, the diskpart utility should be done to get the expanded disk space added to the current disk space of the Disk drive. In Windows Server 2003 disk management, you notice that the added disk space is not yet part of the current disk space (Figure 6-101).

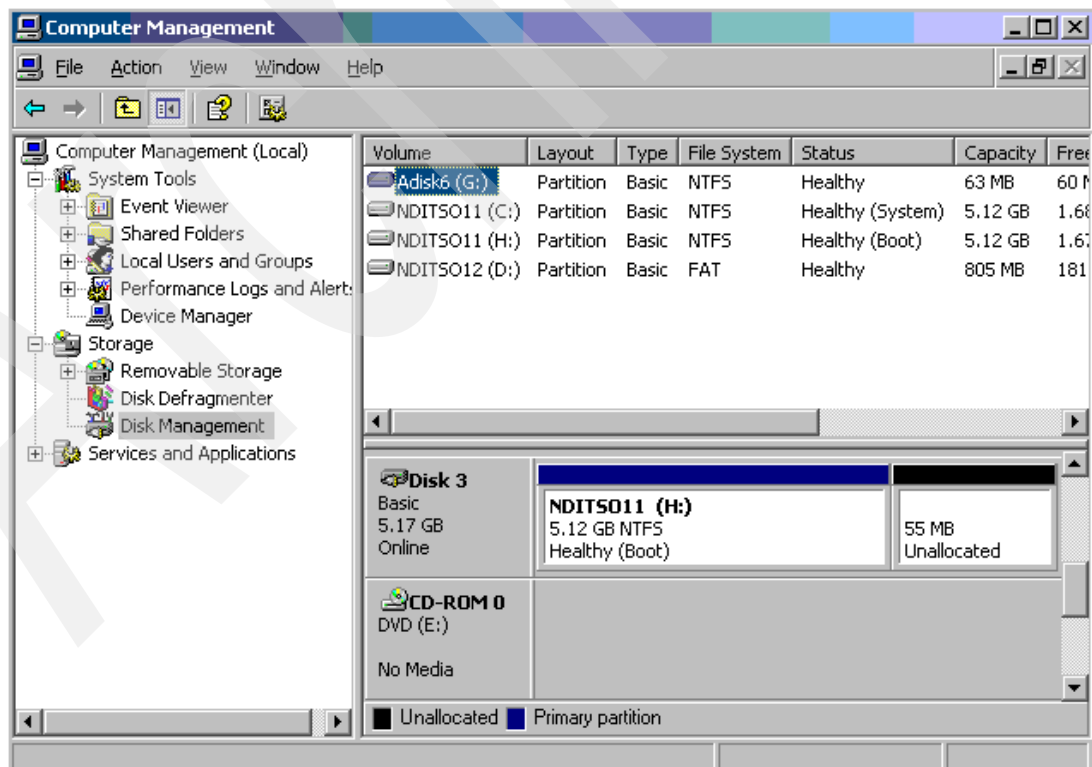


Figure 6-101 Disk Management within Windows Server 2003

9. Follow the steps to perform Diskpart:
 - a. Open a DOS prompt by selecting a shortcut or **Start** → **Run**, and enter CMD.
 - b. On the command prompt, type **Diskpart** and press Enter, the prompt changes from C:\...\> to DISKPART>, and you are now in the Diskpart utility.
 - c. Type **list volume**, a list with volumes appears. You see all the drives and their drive letters. See Figure 6-102.
 - d. Select the new created volume by typing **select volume x**, where x is the volume number. A message returns saying that the volume is selected.
 - e. Type **extend** to expand the drive with the added disk space. It returns with the message “DiskPart successfully extended the volume”.

```
C:\Documents and Settings\Administrator>diskpart
Microsoft DiskPart version 5.2.3790.1830
Copyright (C) 1999-2001 Microsoft Corporation.
On computer: NDITS01

DISKPART> list volume
```

Volume ###	Ltr	Label	Fs	Type	Size	Status	Info
Volume 0	E			DUD-ROM	0 B	Healthy	
Volume 1	F			DUD-ROM	0 B	Healthy	
Volume 2	C	NDITS011	NTFS	Partition	4997 MB	Healthy	Boot
Volume 3	D	NDITS012	FAT	Partition	805 MB	Healthy	
Volume 4	G	NDITS011	NTFS	Partition	4997 MB	Healthy	

```
DISKPART> select volume 4
Volume 4 is the selected volume.
DISKPART> extend
DiskPart successfully extended the volume.
DISKPART> _
```

Figure 6-102 Diskpart in DOS Window

10. Now that the new system drive is extended, we need to unlink the “old” system drive and link the “new” one. For this, because this concerns the system drive, the server should be shut down. Shutting down the server is described in 6.9.3, “Stopping an iSCSI integrated server using CL commands” on page 235.
11. Type **WRKNWSSTG** if not already on this display. Scroll to the new created system disk and put option 11 (Remove link) on the option column in front of the “old” system drive and the “new” system drive and press Enter twice.
12. On the Work with Network Server Storage Spaces (WRKNWSSTG) display, type 10 (Add link) on the option column in front of the “New” system drive, and press Enter.
13. The Add Server Storage Link (ADDNWSSTGL) display appears (Figure 6-103 on page 250), type **NWSD** name for **Network server description**, Type ***Yes** for **Dynamic storage link (DYNAMIC)** and press Enter. Three other parameters show. Type **1** for **Drive sequence number (DRVSEQNBR)** because it is the system drive, which needs to be 1.

Add Server Storage Link (ADDNWSSTGL)		
Type choices, press Enter.		
Network server storage space . . . NWSSTG	>	NDITS01N1
Network server description . . . NWSD	>	NDITS01
Dynamic storage link DYNAMIC	>	*YES
Access ACCESS		*UPDATE
Drive sequence number DRVSEQNBR	>	1
Storage path number STGPTHNBR		*DFTSTGPTH
Bottom		
F3=Exit F4=Prompt F5=Refresh F12=Cancel F13=How to use this display		
F24=More keys		

Figure 6-103 Add Server Storage Link display

14. Start the integrated server described in 6.8.5, “Starting iSCSI integrated server using CL command” on page 228. It now uses the expanded system drive.

6.11.5 Expanding a system drive (C:) having more integrated servers

To expand a system drive (C:) having more than one integrated server is almost the same procedure as having only one integrated server. The difference is that you do not have to make a copy of the “old” system drive. The order to do this is as follows, for details about how to do the steps, refer to 6.11.4, “Expanding system drive (C:) using CL command” on page 246:

1. Shut down the server for which you want to expand the system drive (C:) described in 6.9.3, “Stopping an iSCSI integrated server using CL commands” on page 235.
2. Unlink the “old” system drive (C:) described in 6.11.4, “Expanding system drive (C:) using CL command” on page 246, step 11, only now it needs to be done only for the current system drive.
3. Expand the “old” system drive (C:) with the new capacity as described in 6.11.4, “Expanding system drive (C:) using CL command” on page 246, step 4.
4. Type option 10 (Add link) on the option column in front of the “old” system drive on the Work with Network Server Storage Spaces display by running the command WRKNWSSTG. The Add Server Storage Link (ADDNWSSTGL) display appears. Type NWSD name of the other integrated server for **Network server description**. This server can be active. Type *Yes for **Dynamic storage link (DYNAMIC)** and press Enter. Three other parameters show. Leave these three to the defaults and press Enter.
5. On this other temporary server, perform the Diskpart utility described in 6.11.4, “Expanding system drive (C:) using CL command” on page 246, step 9.
6. Unlink the “new” expanded system drive from the other temporary server, same as step 2.
7. Link the “new” expanded system drive to the original server described in 6.11.4, “Expanding system drive (C:) using CL command” on page 246, steps 12 and 13.
8. Start the original integrated server as described in 6.8.5, “Starting iSCSI integrated server using CL command” on page 228. It now uses the expanded system drive.

6.11.6 Expand a Disk (non-system) using CL command

When expanding a storage space with CL commands, you have to unlink, expand, and link it again, so this is not done automatically as in iSeries Navigator. During this time, this storage space that you are about to expand is not then available for that period of time:

1. Type WRKNWSSTG on a command line, the Work with Network Server Storage Spaces display appears.
2. Scroll to the disk you want to expand and on the option column, type option 11 (Remove link). The Remove Server Storage Link (RMVNWSSTGL) display appears, and press Enter on this display. A status message appears that the link is removed.
3. Type WRKNWSSTG if not already on this display. Scroll to the unlinked put option 2 (Change) in the option column in front of the unlinked drive and press Enter. The Change NWS Storage Space (CHGNWSSTG) display appears. For **Size** (NWSSIZE), type the new capacity in MBs. And press Enter.
4. Link the expanded drive with option 10 (Add link) on the option column in front of the expanded storage space on the WRKNWSSTG Work with Network Server Storage Spaces display.
5. The Add Server Storage Link (ADDNWSSTGL) display appears. Type for **Network Server Description** (NWSD) the same as where the storage space is linked and for **Dynamic storage link** (Dynamic), type *YES. Three other parameters appear; leave these to the defaults and press Enter. If you want to specify another drive sequence number, change the parameter DRVSEQNBR.
6. On Windows Server 2003, perform the diskpart utility to get the expanded disk space added to the current disk space of the Disk drive. In Windows Server 2003 disk management, you notice that the added disk space is not yet part of the current disk space. Follow the steps to perform Diskpart:
 - a. Open a DOS prompt by selecting a shortcut or **Start** → **Run** and enter CMD.
 - b. On the command prompt, type Diskpart and press Enter; the prompt changes from C:\...> to DISKPART>, you are now in the Diskpart utility.
 - c. Type list volume, a list with volumes appears. You see all the drives and their drive letters.
 - d. Select the new created volume by typing select volume x, where x is the volume number. A message returns saying the volume is selected.
 - e. Type extend to expand the drive with the added disk space. It returns with the message, "DiskPart successfully extended the volume."
7. The Disk drive can now be used with the expanded disk space.

6.11.7 Add (link) disk drive using iSeries Navigator

To link a storage space (disk drive), follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks** or **YourSystem** → **Integrated Server Administration** → **Servers** → **YourServer** and click **Linked Virtual Disks**.
2. When you have clicked on **All Virtual Disks**, you see all your disks for all servers configured. So, before doing the link, make sure you have the right one. Right-click the disk you want linked and select **Add link**.
3. An Add Link To Server window opens on which you can select the following using the Pull-down list:
 - a. **Link to server:** Server where this drive should be linked to.

- b. **Link type:** Dynamic (with iSCSI, this is the only option, fixed can be used).
 - c. **Link sequence position:** Select the sequence number for the Disk drive (by default the next available one is used).
 - d. **Storage path:** Specify which storage to be used by this drive.
 - e. **Access to disk drive:** Specify the type of access that servers linked to the disk drive have to the data on the disk drive, default is Exclusive - Update.
4. Select **OK**. The disk drive is linked to the specified integrated server.

6.11.8 Copy a storage space using iSeries Navigator

To copy a storage space (disk drive), follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks** or **YourSystem** → **Integrated Server Administration** → **Servers** → **YourServer** and click **Linked Virtual Disks**.
2. When you have clicked on **All Virtual Disks**, you see all your disks for all servers configured. So, before doing the copy, make sure you have the right one. Right-click the disk you want to copy. Consider the percentage of the disk drive that you are using as the “based on.” Select **New Based On**.
3. The New Disk Based On window opens, type and select the following, and select **OK**:
 - a. Enter a new Disk drive name and a new Description for the new Disk drive.
 - b. If you want to copy the data from the disk drive you selected “New Based On” on, select the **Initialize disk with data from another disk**. When this option is selected, you have to specify the source disk. This is the default value; otherwise, deselect this option.
 - c. Specify the **Capacity**. The capacity can be the same size as or larger than the disk drive you copy from. When you selected the Initialize disk with data from another disk option, you must consider the percentage of the disk drive that was used.
 - d. Specify the **Disk Pool** to be used. Disk Pool 1 represent the Base Pool.
 - e. Specify the **Planned file system** to be used by the new disk drive. It defaults to the file system of the disk drive being copied. NTFS is the most used, except for the D: drive, this one should be FAT.
 - f. You can select **Link disk to server** to link the disk drive to the integrated server of your choice with some other options. These can be selected using the pull-down list:
 - i. **Link to server:** Server where this drive should be linked to.
 - ii. **Link type:** Dynamic (with iSCSI, this is the only option, fixed can be used).
 - iii. **Link sequence position:** Select the sequence number for the Disk drive (by default the next available one is used).
 - iv. **Storage path:** Specify which storage to be used by this drive.
 - v. **Access to disk drive:** Specify the type of access servers, which are linked to the disk drive, have to the data on the disk drive, default is Exclusive - Update.
 - g. Select **OK**.
 - h. A status window opens saying “Creating Disk”. When this finishes, a “Disk must be formatted” window opens, saying that it must be formatted on the server before it can be used. This is only valid when you have not selected Initialize disk with data from another disk.

6.11.9 Unlink disk drive using iSeries Navigator

To unlink a storage space (disk drive), follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks** or **YourSystem** → **Integrated Server Administration** → **Servers** → **YourServer** and click **Linked Virtual Disks**.
2. When you have clicked **All Virtual Disks**, you see all your disks for all servers configured. So, before doing the unlink, make sure you have the right one. Right-click the disk you want unlinked, and select **Remove link**.
3. A Remove Link from Server window opens (Figure 6-104), which shows which server it is linked to and if this server is started. Because this pertains to iSCSI, all drives are dynamically linked so this is no problem. Select **Remove** on this window.

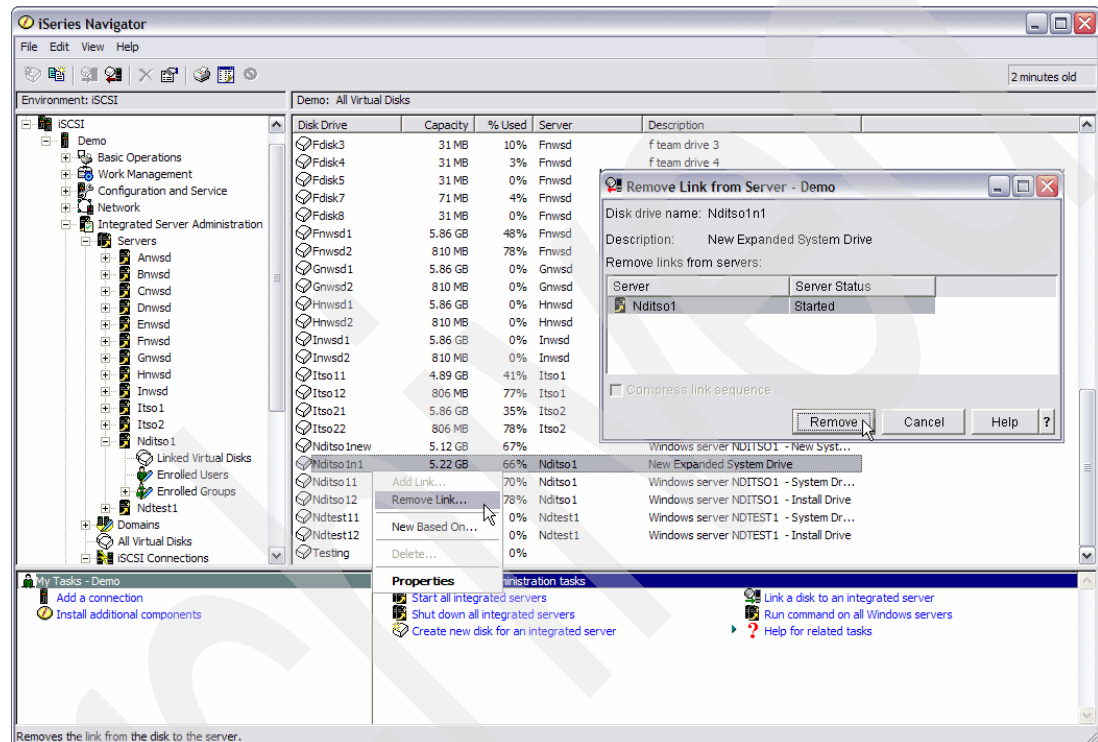


Figure 6-104 Remove Link in iSeries Navigator

6.11.10 Delete disk drive using iSeries Navigator

A storage space (Disk Drive), which needs to be deleted, must be unlinked to any integrated server.

To delete a storage space (disk drive), follow the steps:

1. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks**.
2. When you have clicked **All Virtual Disks**, you see all your disks for all servers configured. So, before doing the delete, make sure the disk drive is not linked to an integrated server. Right-click the disk you want to delete, and select **Delete**.
3. The Confirm Delete Disk Drives window opens. On this window, click **Delete** to confirm the deletion.

6.12 Change Network Host Server Adapter procedure

If you have more than one Network Host Server Adapter (NWSH), you are able to change a specific Server storage space or Virtual Ethernet connection to use another NWSH than the default one or the one which is used by the system and install drive. The NWSH used for the system drive and install drive must be the same one, you cannot separate these.

For a server storage space, the change of the storage paths is done during the Add link function of the server storage space.

For the Virtual Ethernet connection, this is done by changing the attributes of the NWSD for the Virtual Ethernet line.

6.12.1 Change NWSH for server storage spaces using iSeries Navigator

Assume you have a second iSCSI Host Adapter installed on your iSeries and you want to move, for instance, your application data, which resides on the E: drive to this new path. Follow the steps:

1. First you have to add this new iSCSI Host Adapter to the NWSD:
 - a. Shut down the iSCSI integrated server as described in 6.9.2, “Stop/Restart iSCSI integrated server using iSeries Navigator” on page 233.
 - b. Expand **YourSystem** → **Integrated Server Administration** → **Servers** or Click **Servers**.
 - c. Right-click **iSCSI server** to add the NWSH and select **Properties**.
 - d. On the Properties window, select the **Storage Paths** tab and click the right side **Add**.
 - e. The Storage Paths Properties window opens. Select the **Network Host Server adapter** device description using this new added iSCSI Host adapter. The other fields can be left to the defaults. IP security rules are not supported at the time this book was written. And click **OK**. Click **OK** again on the NWSD properties window.
2. Expand **YourSystem** → **Integrated Server Administration** → **All Virtual Disks**.
3. Remember to notice the drive number before removing the link. Right-click a **drive**, which you want to move from the “old” storage path to the “new” storage path and select **Remove Link**. This process is described in 6.11.9, “Unlink disk drive using iSeries Navigator” on page 252.
4. Right-click the same **drive** again and select **Add Link** with the following selections from the pull-down list and click **OK**:
 - a. **Link to server**: Same server as the drive was linked to.
 - b. **Link type**: Dynamic (default)
 - c. **Link sequence position**: Select the same number as it was.
 - d. **Storage path**: New storage path.
 - e. **Access to disk drive**: Exclusive - Update (default)
5. Start the iSCSI integrated server as described in 6.8.4, “Starting iSCSI integrated server from iSeries Navigator” on page 224. The disk drive now uses the new storage path.

6.12.2 Change NWSH for Virtual Ethernet connection

Chapter 8, “Virtual Ethernet LAN” on page 305, describes this topic.

Backup and Recovery

This chapter describes backup and recovery of Windows running on integrated xSeries hardware from both the System i5 and Windows perspectives. Because Windows running on integrated xSeries hardware combines two operating systems (Windows and i5/OS), you can choose to manage backups by using either i5/OS commands or Windows backup applications, or a combination of both. We concentrate on the i5/OS-centric view, because Windows native backup and recovery is largely specific to the backup application you use. These applications are well documented in their own right.

Important: This chapter is specifically written for i5/OS V5R4, System i5, and iSCSI.

7.1 Overview of backup and recovery

In this section, we introduce and position the different methods of saving and restoring Windows data, and we give recommendations about the way we think the backups should be done. We also provide tips about improving the performance and availability of your Windows servers.

You should understand the concepts that are involved with backup and recovery before you plan your backup strategy. Therefore, read the entire chapter before you start working on your plan.

Because the integrated Windows server environment is a combination of two operating systems, i5/OS and Windows, there are several options for saving and restoring data from which you can choose. The types of backup you can perform, and therefore, the options you have for restoring data are different, depending on whether you are looking at backup from an i5/OS or Windows perspective. You might choose to manage your backups using i5/OS functions, Windows functions, or a combination of both. It depends on your skills, what you are backing up, how much time you have to perform the backup, and how you want to restore the data. Both the i5/OS and Windows operating systems have their strengths and limitations, so understand what each system does well and plan your strategy around those strengths.

Terminology:

In this chapter, all references to Windows apply to Windows Server 2003 only, because this is the only version of Windows that is supported in the iSCSI environment.

We use the term *storage space* to mean a network server storage space. A storage space is a chunk of System i5 single level disk storage that Windows sees as a physical disk drive. We use the terms *storage space* and *Windows drive* synonymously.

An *integrated Windows server* is an instance of Windows running on an xSeries server or BladeCenter Blade. Each integrated Windows server has its own corresponding network server description (NWSD). We also refer to integrated Windows servers as simply *Windows servers*.

If you have a Windows background, we encourage you to review your previous backup strategy and planning in light of what you read in this chapter. Windows running on Integrated xSeries Server and Integrated xSeries Adapter effectively acts as a guest operating system under i5/OS, which means more flexibility and new techniques that you can use to save your Windows files. Using the techniques described in this chapter, you might be able to save your Windows files more quickly and efficiently than if Windows were running on a stand-alone server. You might also be able to recover from a Windows failure that would otherwise require you to rebuild the server.

In terms of backup and recovery, there are some major advantages of running Windows on Integrated xSeries Server and Integrated xSeries Adapter compared with running Windows on a stand-alone server. These include the capability to:

- ▶ Back up a *complete* Windows server to tape or disk so that it can be easily restored in the event the Windows server fails and cannot be restarted. Backing up and recovering a complete Windows image is very difficult to do on a stand-alone Windows server.
- ▶ Create “online” backups of Windows drives on System i5 disk storage, which can be linked to, and accessed by, a Windows server for fast recovery of individual files.

- Use sophisticated products such as Backup Recovery and Media Services (BRMS) to provide centralized, unattended backup of Windows servers, and management of tape media.
- Incorporate the backup of a Windows server into an i5/OS backup procedure using native i5/OS functions, or the advanced functions of a product such as BRMS. This enables you to save both the i5/OS and Windows environments in one operation, and take advantage of the high-speed tape drives and powerful software support that the System i5 offers.
- Use high-speed System i5 attached tape drives to back up Windows files using Windows-based utilities and backup applications. This capability is referred to as *virtual tape support*.

Never underestimate the importance of backup. No matter how much it costs, it almost always costs less to your business than trying to manually recover or recreate lost data. Always test your backups to make sure that you can restore them successfully.

For detailed information about i5/OS backup and recovery, we recommend that you refer to *System i5 Backup and Recovery*, SC41-5304, and the Information Center at Web site:

<http://publib.boulder.ibm.com/infocenter/iseres/v5r4/topic/books/sc415304.pdf>

7.1.1 Backup from a server-centric perspective

The backup of Windows files can be performed from either the i5/OS or Windows side. We term this i5/OS-centric and Windows-centric as shown in Figure 7-1.

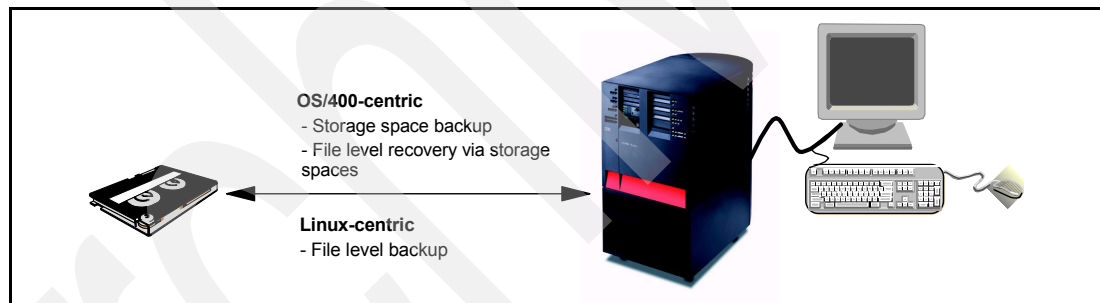


Figure 7-1 Backup from a server-centric perspective

We recommend that you use a combination of storage space and file level backup functions to obtain the best overall save-and-restore strategy that satisfies your business requirements.

i5/OS-centric backup

i5/OS-centric backup is synonymous with using i5/OS commands to back up Windows data at a storage space level. This type of backup is ideal for the recovery of Windows Servers because it is quick and easy to back up a complete server. We call this *storage space backup* and restore. The objective of storage space backup and restore is to recover a complete Windows disk drive, or even a complete integrated Windows server, after a catastrophic failure.

Although you cannot restore individual files directly from a storage space using i5/OS commands, it is possible to recover a single Windows file from a storage space using a technique known as *file level recovery* via storage spaces. This technique enables you to recover individual Windows files without needing to use Windows backup and restore utilities. Therefore, you can greatly simplify your Windows backup and recovery strategy by using storage space backup for both your disaster recovery and file level backups.

We discuss both storage space backup and file level recovery via storage spaces in 7.3, “i5/OS-centric backup” on page 265, and 7.4, “i5/OS-centric recovery” on page 278.

Windows-centric backup

Windows backup utilities and applications provide file level save and restore capability for Windows servers. This method might be preferred by customers migrating from stand-alone Windows servers, who prefer to save their files using a Windows-based backup application, and already have the infrastructure in place.

If you want to pursue a Windows-centric backup and recovery strategy, there are a number of Windows utilities and applications that you can use. Because backing up and restoring files using a Windows backup application is not specific to the Windows integration support, we do not describe how to use Windows backup applications in detail. We give an overview of the backup utilities included with the supported distributions of Windows, and other commercially available Windows backup applications.

We discuss Windows file level backup and recovery in 7.5, “Windows-centric backup and recovery” on page 290.

7.2 Planning a backup strategy

The objective of this section is to help you understand the special characteristics of Windows running on the Integrated xSeries Server or Integrated xSeries Adapter, and provide input to your overall backup strategy.

We do not attempt to help you define your backup strategy in detail, or cover how to back up the System i5 as a whole. Both of these topics are well documented elsewhere. We do recommend that you consult the manual *System i5 Backup and Recovery*, SC41-5304, and the Information Center at Web site:

<http://publib.boulder.ibm.com/series/>

7.2.1 Staged backup

An important consideration when planning a backup strategy is the use of staged backup. With a staged backup, you first back up data to disk and then (optionally) save this intermediate backup to tape as shown in Figure 7-2.

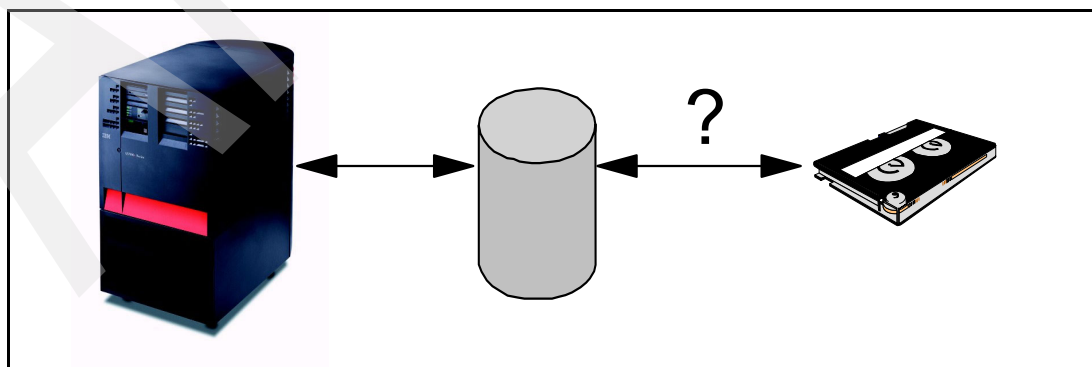


Figure 7-2 Staged backup

Because saving to disk is usually faster than backing up directly to tape, it can be very useful, even mandatory, to first save your data to disk when your backup window is small. A backup

window is defined as the time that is available to perform a backup of your data, which usually corresponds to the length of time that the Windows server can be made unavailable to users. As soon as the backup to disk has been completed, the server can again be made available to users. The backup on disk can then be saved to tape at a later time.

In the case of i5/OS objects, saving them to disk on the System i5 can be accomplished by using an object called a *save file*. A save file acts as a common receptacle for saving different types of objects such as libraries, files, and storage spaces. You can use the same i5/OS commands to save to a save file as you would use to save directly to tape. Instead of specifying a tape device name, you specify *SAVF. You are also prompted for the name of the save file and the library where it is to be stored.

Just as you can back up to disk, you can also restore from disk. This technique provides a very fast method of restoring data, especially because you do not need to waste time locating and mounting a tape.

The downside of saving to disk is the cost of the additional disk storage required. However, the cost depends on whether you subsequently save the data to tape and delete the disk backup, or leave the disk backup in place for fast retrievals.

Note that staged backup is complementary to tape backup; it does *not* replace tape backup. You always need to ultimately save your data to tape in case you lose your system and you cannot restore from disk.

Staged backup can be performed from both i5/OS and Windows as follows:

► **i5/OS**

A storage space can be saved to disk either by saving it to a save file, or by copying it to a another storage space.

► **Windows**

Windows files can be saved to disk through the Windows console using Windows utilities and backup applications. Windows backup utilities, such as tar, cpio, and dump, provide a capability similar to that provided by i5/OS save files in that you can save multiple files to a single archive on disk or tape. Archives can also be created directly on a shared IFS directory using System i5 NetServer™ and Samba.

We discuss the various methods you can use to perform staged backup later in this chapter.

7.2.2 Backup and recovery tips

As you read this chapter, you will come to your own conclusions about the options that are most suited to your own environment. Here we provide some tips that you should consider when formulating a backup strategy:

- Before you decide how you are going to back up your system, you must first decide how you want to restore it; that is, at a storage space level, file level, or both.
- Next, decide whether you are going to use an i5/OS-centric or Windows-centric approach, or a combination of both. Keep in mind that backing up data at a storage space level can only be performed from i5/OS.
- Importantly, you need to calculate how much data to save and how frequently. Many of the i5/OS objects that comprise the integrated Windows server (such as the network server description) do not change on a regular basis. Therefore, it is pointless to save these objects every day. Conversely, you might need to save user files daily.
- Determine your backup window. Backups in a production environment usually require that all user applications be shut down because you can strike problems with data integrity if

users are writing to files on the Windows server while they are being backed up. Incomplete transaction data might be saved which could cause problems should you need to restore the data. The *backup window* is the length of time that the server can be made unavailable to users to avoid these issues.

- ▶ Once you decide how much data to save, and how long your backup window is, you can decide whether to take the staged approach and save some of your data to disk before backing up to tape. You would normally only take this approach if your backup window is not long enough to be able to save the data to tape.
- ▶ Before you start backing up a Windows server, the network server description must be in the correct state, depending on the type of backup you are performing:
 - To perform an i5/OS storage space backup, the integrated Windows server must be shut down.
 - To perform a Windows file level backup, the integrated Windows server must be started.
- ▶ If possible, do not store production data on the system drive storage space. This enables you to back up the storage space where the Windows operating system files are stored, independently of applications and data.
- ▶ Try to keep application software on a drive or drives separate from the system and data. Applications rarely change and do not require frequent backups, so you can eliminate these storage spaces from your daily routine.
- ▶ Generally speaking, try to keep static data and frequently modified data on separate drives. Static data changes infrequently and only needs backing up when it changes, whereas frequently modified data needs to be saved regularly.
- ▶ Remember that all data on System i5 disk storage is scattered across every physical disk drive in the Auxiliary Storage Pool (ASP). This means that the size of the storage spaces you create, which Windows sees as disk drives, has absolutely no impact on performance. Therefore, you can create very large storage spaces of up to 64 GB at V5R2 and 1 TB at V5R3 to hold large amounts of data, without adversely affecting performance. This might simplify your backup strategy.
- ▶ Always save the complete integrated Windows server, including non-storage space components such as the network server description and communications objects, as soon as you are satisfied that the server is installed and configured correctly. This backup enables you to recover a working server should you subsequently have an unrecoverable failure of the Windows operating system. Ensure that these non-storage space objects are included in a periodic, complete i5/OS backup.
- ▶ Make frequent backups of the system drive storage space where the Windows operating system is stored. If you experience an unrecoverable failure of the Windows operating system, you can recover by simply restoring the system drive from backup. In this case, you want a recent copy of the system drive to recover as many changes to the Windows operating system as possible. These changes could include patches and new user accounts.
- ▶ The fastest way to save the system drive (or any other drive for that matter) is to use i5/OS storage space backup to save an image of the storage space that corresponds to the Windows drive where the data is stored.
- ▶ If you have already implemented a backup strategy using Windows backup utilities, you might want to continue using the backup infrastructure you already have in place. However, we strongly recommend that you also consider using i5/OS storage space backup to complement your current procedures.
- ▶ If you are new to Windows, you probably have an i5/OS-centric backup strategy that is well established. We recommend that you review this strategy to verify that the integrated

Windows server components are included. Make sure you also consider implementing a file level backup strategy for your integrated Windows server using one of the utilities included with the supported Red Hat and SUSE distributions, or some other commercially available backup application. Review these to see which best suits your needs. We recommend that you consider IBM Tivoli® Storage Manager for larger installations, or the Windows backup utilities for smaller shops.

- ▶ If you need disk protection (RAID-5 or mirroring), implement this at the i5/OS level and do not use any form of Windows-based protection. Using Windows software RAID-5 or RAID-1 (mirroring) would adversely affect performance and add no additional protection whatsoever. This is because the drives that Windows sees are in fact virtual disks created from the System i5 disk pool. Each virtual disk is overlaid on top of all the physical disk drives in the pool. Therefore, if you implement Windows software RAID-5 or RAID-1, you are simply duplicating the System i5 native RAID-5 or mirroring protection.
- ▶ If you are restoring an integrated Windows server to a different System i5 system or i5/OS partition, or you want to bring up a particular instance of Windows on another Integrated xSeries Server or Integrated xSeries Adapter in the same i5/OS partition, remember to check the hardware resource name of the target Integrated xSeries Server or Integrated xSeries Adapter. The resource name is probably different than the source, and you would need to modify the network server description accordingly.

7.2.3 Automating backup and recovery

Backup is one of the key areas that many people want to automate from i5/OS because of the sophisticated backup application support, high speed tape drives, and powerful scripting capabilities. Therefore, when we talk about automation, we are really looking at automating backup and recovery-related functions from the i5/OS side. Typically, most people want to automate backup by running unattended backups overnight when their systems are not being used. You can automate the backup (and recovery) of an integrated Windows server from the i5/OS side at a storage space level. This chapter provides some examples of how you can write CL programs to automate backup and recovery tasks.

If you want to perform an unattended storage space backup of an integrated Windows server, you can incorporate the backup into a program by including the appropriate CL commands. This means that you can save both your i5/OS and Windows environments unattended, as part of a single backup procedure.

7.2.4 Backup technique positioning and recommendations

When selecting a backup technique, first decide whether you want to save from the i5/OS side, Windows side, or a combination of both. Each technique has its own strengths and limitations. If you come from a Windows background, you might gravitate toward a Windows-centric backup strategy. However, you should consider performing i5/OS storage space backups, even if you choose to save at a file level from the Windows side. There is no real equivalent to an i5/OS storage space backup in Windows.

If you decide to go with the i5/OS backup option, we recommend that you review your Windows backup requirements with the person in your organization who is responsible for backing up the System i5 to make sure that the i5/OS backup procedures include the Windows integration components (network server description, communications objects, and so on). Many organizations modify the default i5/OS backup options to suit their own requirements. Verify that your organization has not removed the commands that save the Windows integration objects.

Table 7-1 provides a high-level positioning guide for each of the tape backup techniques described in this chapter. Each backup technique is rated as High, Medium, or Low against each positioning attribute.

Table 7-1 Positioning tape backup techniques

Tape backup technique	Performance	Function	Cost
i5/OS storage space backup	H	H	L (free)
Windows file level backup using utilities supplied with the supported Red Hat and SUSE distributions	H	M	L (free)
Windows file level backup using commercially available backup applications, such as Tivoli Storage Manager	H	H	M to H

Recommendation: To be fully protected against data loss, we recommend that you perform backups at both a storage space and file level. Storage space backup protects you against a complete loss of the System i5 system or Windows server, whereas file level backup enables you to recover individual files.

Therefore, you should perform both of the following backups to be fully protected:

1. Storage space backup

This is a backup of all Windows disk drives storage spaces on a regular basis, for example, once per week. The only way to perform a storage space backup is from i5/OS. Such a backup can easily be incorporated into a System i5 backup plan.

2. File level backup

This is a backup of Windows files on a regular basis, such as every day for production files or once per week for system files. You would normally perform this type of backup from Windows using a Windows backup utility or some other backup application.

Note that you do not need to back up at a file level in order to restore individual files. It is possible to recover individual files from a storage space by mounting the storage space as a new Windows drive and copying files from the newly linked storage space to their original locations within the Windows file system.

As part of a file level backup you should also perform Incremental or differential file level backups of your production data. This is a backup of files that have changed since the last complete backup. You should perform an incremental or differential backup on a frequent basis, probably daily.

7.2.5 Recommended backup schedule

As part of your backup strategy, you should have a backup schedule. Such a schedule tells you when to save the various components of the Windows integration environment.

To minimize recovery times, the more volatile the data is (that is, the more often it changes), the more regularly it should be backed up.

We recommend that you save integrated Windows servers at both a storage space and file level. This protects against both the loss of the System i5 or Windows server, and the loss or corruption of an individual file. We also recommend that you save your critical Windows system files regularly. We define a critical Windows system file as one that changes on a regular basis and contains information vital to the operation of the Windows server. You can

save these critical files either as part of an i5/OS storage space backup, or individually as part of a Windows file level backup.

Table 7-2 summarizes the recommended backup schedule for components of the Windows integration environment. Use this table as a guide to create your own backup schedule.

Table 7-2 Recommended backup schedule

Time frame	i5/OS integration objects ¹	Storage space backup			File level backup	
		Windows system and installation drives	Windows application program drives ²	Windows data drives	Windows volatile user data ³	Windows non-volatile user data ³
Daily				X	X ⁵	
Weekly		System drive		X	X ⁶	X
After installation	X	System and installation drives				
After making changes to i5/OS integration objects	X					
After making changes to the Windows operating system		System drive				
After installing or updating a Windows application ⁴		System drive	X			
After installing an integration service pack on Windows ⁷	X	System and installation drives				

¹i5/OS integration objects include the network server description and communication objects.

²Windows application program drives include all Windows drives where user applications are installed. We refer to this as application data as distinct from user data.

³Windows volatile user data is data that is changing on a daily basis, as opposed to Windows non-volatile user data, which refers to Windows files that change infrequently. We draw a distinction between user data and application data. Volatile data includes critical Windows files.

⁴Installing a new Windows application or applying patches can update critical Windows files, and create new files on the system drive. Therefore, you need to save the system drive plus any other drives containing Windows applications. It is always a good idea to store applications and data on different drives.

⁵Perform a daily incremental or differential backup to save files that have changed since the last (weekly) full file level backup.

⁶Perform a full file level backup, for example, on a weekly basis.

⁷This function is not available in the initial implementation of Windows integration support.

7.2.6 Hot spare

One of the unique features of the System i5 Integration for Windows Server architecture is the concept of separating a Windows server instance from the hardware it is running on. This concept is equally applicable to Windows running on integrated hardware.

When you view integrated Windows and Windows servers through System i5 Navigator, as shown in Figure 7-3, each icon in the Windows Server column represents an *instance* of Windows or Windows server, *not* a piece of physical integrated hardware (Integrated xSeries Server or Integrated xSeries Adapter). For example, all three servers, Redhat2, Suse8, and Xw2003a, use the same hardware resource, Lin02 2689-001, which is an Integrated xSeries Adapter attached xSeries machine. This means that all three server instances were created on this same integrated hardware resource. Note, however, that only one server instance can run on a particular integrated hardware resource at a time. Therefore, while Suse8 is running on resource Lin02, Redhat2 and Xw2003a are “dormant”.

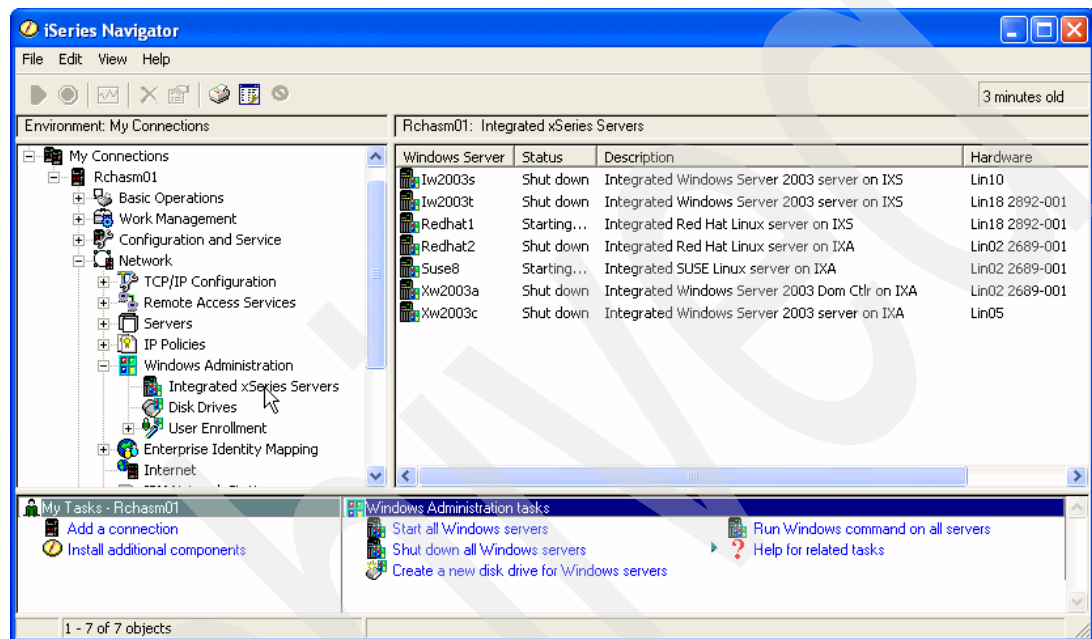


Figure 7-3 Windows and Windows server instances

This abstraction of server instances from the hardware they run on means that you can point a server instance to another piece of physical integrated hardware and start the server on the new piece of hardware. You can do this by simply changing the resource name in the server's network server description (NWS). Of course, because of the wide variation in the hardware design of Intel-compatible machines, there are restrictions on the types of hardware that you can switch a server instance to. These restrictions are caused by the fact that different pieces of Intel-compatible hardware require different drivers. Unless the drivers that run on the source hardware you are switching from, are compatible with the drivers on the target hardware you are switching to, the server instance might not start on the target machine. For a more detailed discussion of driver compatibility, refer to 7.4.1, “Integrated hardware considerations” on page 279.

Assuming you have compatible pieces of integrated hardware, it is a simple matter to switch an instance of Windows or Windows server from one piece of hardware to another by changing the resource name in the server instance's NWS.

Given the separation of integrated server instances from the hardware they run on, and the ease of switching a server instance from one piece of integrated hardware to another (compatible) piece of hardware, you can use these functions to implement a *hot spare* capability for your integrated Windows and Windows servers. As shown in Figure 7-4 on page 265, you can reserve one of your Integrated xSeries Servers or Integrated xSeries Adapter attached xSeries as a hot spare. Should an integrated hardware resource fail for

some reason (for example, a failed processor or power supply), all you need to do to get the server instance up and running again is to:

1. Shut down the server instance running on the failed hardware through System i5 Navigator or i5/OS green screen.
2. Change the resource name in the server instance's NWSD to the hot spare machine.
3. Restart the server instance.

This hot spare capability can be thought of as *hardware hot spare*. It is a very simple and effective way to protect your integrated Windows and Windows servers from hardware failures. This is in contrast to situations where servers are not integrated with the System i5 where the most common way to recover from a catastrophic hardware failure is to rebuild the server, a potentially very time consuming and complex operation.

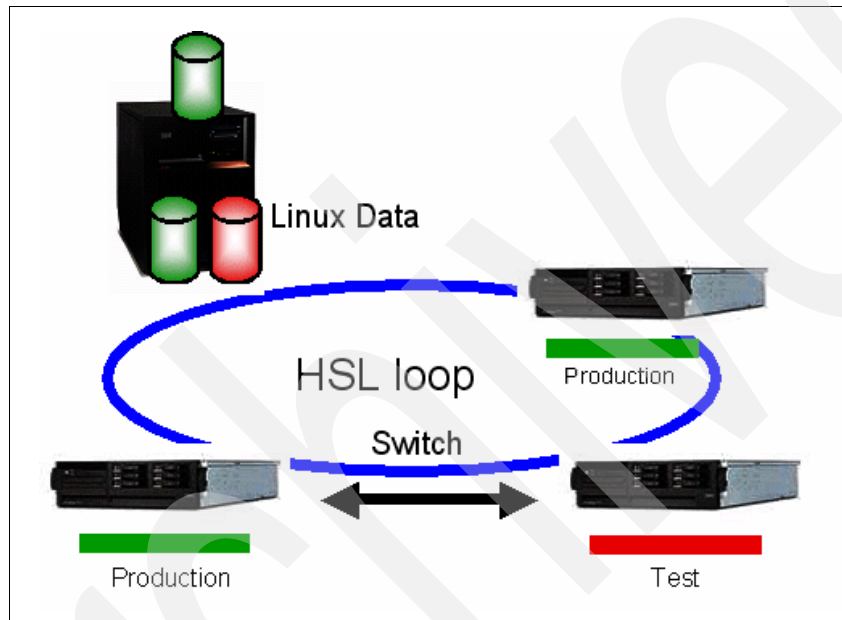


Figure 7-4 Switching a server instance to a hot spare

It should be noted that the hot spare capability we have been describing does not protect you against a software failure that renders your server inoperable. Such a failure could be caused, for example, by a faulty driver or bad application. Again, with a stand-alone server you would probably need to rebuild the server. However, with an integrated server you can save and restore the server's drives as complete objects. Therefore, even if you experience a catastrophic failure of the integrated server's system drive, you should be able to recover by simply restoring a copy of the drive from a recent backup. This capability can be thought of as *software hot spare*.

Both hardware hot spare and software hot spare are simple but effective techniques that provide your integrated Windows and Windows servers with a high level of availability.

7.3 i5/OS-centric backup

We define i5/OS-centric backup, as it relates to running Windows on the Integrated xSeries Server or Integrated xSeries Adapter, as the capability to save the objects that comprise integrated Windows servers by using i5/OS commands.

A number of objects are created on i5/OS as a result of installing the Windows integration software. These objects include:

- ▶ Network server description (NWSD)
- ▶ System and installation storage spaces
- ▶ Virtual Ethernet line descriptions

To perform a disaster recovery backup, you need to save all these objects if you want to recover properly. You should plan to take a complete backup of your Windows environment so that it can be restored from scratch if necessary. Make sure you have a plan to restore on a different System i5 server, possibly at a different site. This should enable the integrated Windows server to be up and running fairly quickly, depending on the volume of data to be restored.

From a backup and recovery perspective, the most important component of an integrated Windows server is its storage spaces. In addition to the system and installation storage spaces, you can create other storage spaces for your server after it has been created. By backing up these storage spaces on a regular basis, you can completely recover a failed Windows server by simply restoring the storage spaces, which comprise the server's drives. We, therefore, cover the backup and recovery of storage spaces extensively in this chapter.

Important: In terms of i5/OS-centric backup, the key benefit of running Windows on the Integrated xSeries Server or Integrated xSeries Adapter is the ability to quickly and easily save a complete image of an integrated Windows server to tape or disk by backing up its storage spaces. This capability is difficult to implement in a stand-alone Windows environment.

7.3.1 Storage space backup overview

Backups using Windows standard backup utilities, and applications such as Tivoli Storage Manager, are inherently file-oriented in nature. With these applications, you save at a file level and restore at a file level. Because Windows is effectively acting as a guest operating system under i5/OS, you can save i5/OS storage spaces (which are equivalent to Windows drives) as discrete entities. Because i5/OS saves each storage space as a single file in streaming mode, save rates are very high, and you can take a complete copy of all Windows drives, including the system drive, in a short amount of time. However, because you save a storage space as a single entity, you must restore it in its entirety. You cannot directly restore individual files from a storage space. Because a storage space is saved and restored as a single, complete entity, this is true disaster recovery backup.

The objective of a storage space backup is to recover an integrated Windows server drive or drives as fast as possible. Such a situation might arise after a fire or flood when the System i5 system has been totally destroyed, or when the Windows server has become unusable. In this case, you are more concerned with the speed of restoring the Windows server as a whole, rather than restoring individual files.

Storage space backup is fast, and it enables you to incorporate a backup of Windows files into an unattended backup of the hosting i5/OS partition. Because you can save all Windows drives, including the system drive, as complete units, it is quick and easy to recover a failed server by simply restoring its storage spaces. This is a major advantage of using i5/OS storage space backup to perform a save your integrated Windows server, versus trying to use a Windows backup technique to do this.

Important: To perform a storage space backup, the integrated Windows server must be shut down.

7.3.2 Storage space backup tips

Here are some tips that you should keep in mind when performing a storage space backup:

- ▶ You can restore a storage space over the top of an existing storage space without deleting it first.
- ▶ You can save (SAV command) or copy (CRTNWSSTG command) a storage space regardless of whether it is linked to a network server description or not.
- ▶ If you save a storage space (SAV command) that is statically linked (*FIXED) or dynamically linked (*DYNAMIC), it restores in a linked condition provided that you restore the storage space *before* the NWSD. If you save a storage space (SAV command) while it is unlinked, it restores in an unlinked condition.
- ▶ When you save or restore a storage space, you should see a completion message of “*n* objects saved or *n* objects restored”, where *n* can range from 3 to 6. The objects are:
 - The storage space directory object, which has the same name as the storage space.
 - QFPCLTSTG*n* is a stream file object that contains the data. *n* can range from 1 to 4.
 - QFPCONTROL is a stream file object that contains header information.

7.3.3 Components of the Windows integration environment

The Windows integration environment can be broken down into four basic components:

- ▶ i5/OS configuration objects
- ▶ Predefined drives
- ▶ User drives
- ▶ Integration code

These basic components are composed of logical groupings of objects that should be saved together to maintain synchronization between the various objects that comprise the Windows integration environment. The objects that comprise each basic component are:

- ▶ **i5/OS configuration objects**

These include the:

- Network server description (NWSD) that i5/OS uses to control the server and describe its attributes
- Communication lines, controllers, and devices that the integrated Windows server uses to communicate with i5/OS and other servers

Note that you do not need to save controller and device descriptions. If necessary, they are automatically created when the integrated Windows server is started.

We recommend you save these objects after you have:

- Finished installing the integrated Windows server
- Made changes to the network server description
- Made changes to the communication environment

- ▶ **Predefined drives**

When you install Windows on an Integrated xSeries Server or Integrated xSeries Adapter, i5/OS creates the predefined system and installation drives only. You can then create additional drives to store application and user data. The OS/400 storage space objects that correspond to these drives reside in the /QFPNWSSTG directory of the IFS.

The predefined drives are the:

- System drive storage space, which contains the Windows operating system
- Installation drive storage space, which contains the IBM-supplied drivers and utilities

We recommend you save these objects after you have:

- Finished installing Windows on the Integrated xSeries Server or Integrated xSeries Adapter
- Made changes to the Windows operating system and kernel
- Installed or updated a Windows application

► **User drives**

These are drives which can be optionally created after the Windows server has been installed. Typically, these drives would be used to store Windows application and user data:

– **Windows application data**

Windows application data could be stored on:

- System drive
- Other drives where Windows applications are installed

We recommend you save these drives after you have installed or updated a Windows application.

Save all storage spaces that contain application data in the one backup operation to maintain synchronization between the Windows operating system and the applications.

– **Windows user data**

We recommend you save this type data on a regular (either daily or weekly) basis so that you have a backup of all updates to user files. Remember that this is a backup at a storage space level; you cannot directly restore individual files from this backup.

Save all storage spaces that contain user data in the one backup operation to maintain synchronization between files on different Windows drives.

► **Integration code**

The Windows integration code is included in the System i5 Integration for Windows Server program product (5722-WSV *BASE and option 2) and includes:

- QLSVT library
- /QIBM/ProdData/lsvt IFS directory

There is also a component of the integration code that is installed on the Windows server itself to provide drivers and utility functions.

We recommend you save this program product after you have:

- Finished installing Windows on the Integrated xSeries Adapter or Integrated xSeries Server
- Applied PTFs to the integration code

You can save the integration code by using the SAVLICPGM CL command.

To ensure the stability of Windows, be careful if you save any one of the Windows drives without saving the others.

We recommend that you save the system and installation drives frequently. The system drive should be saved at least weekly because it contains critical Windows files that are constantly changing. Instead of saving the system drive to tape, you can back it up to a save file on System i5 disk storage. If you use compression, this does not take a large amount of disk space. Saving to disk gives you the ability to restore the system drive quickly, should you

need to recover it, and get your integrated Windows server up and running again with a minimum of disruption.

There is another component of the integrated Windows server that is not an object, and, therefore, you cannot save it or restore it. TCP/IP interfaces must always be recreated if they are deleted or missing. Refer to “Recreating TCP/IP interfaces” on page 281 for more information.

You might want to save the complete Windows integration environment in order to restore a server on another System i5 after a disaster when you cannot recover your primary system. In this case, you need to save all of the components that we have discussed to enable you to restore the server from scratch. Note that performing a complete system save (option 21) from the i5/OS Save menu saves all components of the Windows integration environment.

7.3.4 Using BRMS to back up an integrated Windows server

IBM Backup, Recovery and Media Services (BRMS), Licensed Program Product 5722-BR1, is the IBM strategic solution for planning and managing the backup of a System i5 server. It provides all of the functions that most System i5 users need to implement a fully automated, single system backup, recovery, and media management strategy.

BRMS facilitates centralized management of media by maintaining a consistent view of removable tape media, its contents, location, and availability across multiple System i5 servers or i5/OS partitions referred to as *networked systems*. This common media scratch pool contains shared tape volumes, which are eligible for use by any participating networked system. When a networked system uses one of the shared volumes, that usage is broadcast to all networked systems so that each system has a current view of the active media and the available expired media.

BRMS provides the System i5 server with support for policy-oriented setup and execution of archive, backup, recovery, and other removable media-related operations. BRMS uses a consistent set of intuitive concepts and operations, which can be used to develop and implement a backup strategy tailored to your business requirements. The user interface is menu-driven, with a significant number of functions enabled through the optional BRMS System i5 Navigator client, a plug-in to System i5 Navigator.

BRMS policy support enables you to automate the backup of storage spaces and other objects, which comprise the Windows integration environment. BRMS can shut down integrated servers prior to backing them up and then restart them after backup has completed. Therefore, when you back up your integrated Windows servers you can exploit the benefits of this powerful, sophisticated product by incorporating the save into a BRMS backup schedule.

The base BRMS product supports an unlimited number of media, shared tape devices, automated taped libraries, and IBM Tivoli Storage Manager servers. It supports backup of a single library or single QSYS.LIB object in parallel using up to 32 tape devices. Parallel backup with its easy-to-use interface lets users shorten backup windows by using more tape devices. Using parallel backup with an automated tape library device, users can save a large library for example, to all currently available tape resources. Parallel backup also supports *ALLUSR, *IBM, and generic library names.

Note: The base BRMS product does not support shared media, archive, dynamic retrieval, or automated migration operations.

For more information about BRMS, refer to the Web site:

<http://www-03.ibm.com/servers/eserver/series/service/brms/>

BRMS optional features

BRMS is structured to allow the addition of functions and features incrementally as business needs change and grow. There are two optional features that can be added to the base product to build a full function BRMS system as explained in the following sections.

BRMS Network Feature

With the BRMS Network Feature, a BRMS system is connected to other BRMS systems in the network using native TCP/IP, Advanced Peer-to-Peer Network (APPN), or both. A BRMS network system shares the inventory and policies associated with media managed by a central BRMS system. In addition, users can view the saved history of any system in the network from a single system. If two systems in the network share a common device, a user can automatically restore objects on one system that are saved on another system.

The networking feature also allows users to off-load their media duplication tasks by using one system in the network to duplicate media on behalf of another system in the network. The systems in a BRMS network can be other System i5 systems or individual i5/OS partitions.

BRMS Advanced feature

The BRMS Advanced feature enables Hierarchical Storage Manager (HSM) archive with HSM dynamic retrieval, and automated auxiliary storage pool (ASP) data migration.

Parallel backup works with the BRMS Advanced feature to allow for parallel archive and parallel dynamic retrieval of a single object. The ability to dynamically retrieve a large database file in parallel helps to reduce the window of the retrieval process. Therefore, it increases the benefits of using HSM archive and dynamic retrieval support.

The BRMS Advanced feature allows archive capabilities of database files, stream files, and documents based on frequency of use, inactivity limit, object size, or ASP thresholds.

7.3.5 i5/OS-centric backup methods

There are three basic methods for performing an i5/OS-centric backup of the Windows integration environment:

- ▶ **Using the Save menu**

Refer to 7.3.6, “i5/OS-centric backup using the Save menu” on page 270.

- ▶ **Using CL commands**

Refer to 7.3.7, “i5/OS-centric backup using CL commands” on page 272.

- ▶ **Using CL programs**

Refer to 7.3.8, “i5/OS-centric backup using CL programs” on page 275.

Note: You cannot perform an i5/OS-centric backup through System i5 Navigator.

7.3.6 i5/OS-centric backup using the Save menu

You can use the Save menu (GO SAVE) to perform an i5/OS-centric backup of the Windows integration environment to tape. The i5/OS commands for saving the system are available on the second panel of the Save menu, as shown in Example 7-1 on page 271.

Example 7-1 Save menu options

Save System and User Data

- 21. Entire system
- 22. System data only
- 23. All user data

To save all components of the Windows integration environment, enter GO SAVE and select option 21. This option backs up the entire i5/OS partition including any integrated Windows servers that are installed in it. Option 22 saves system data only. Option 23 saves all user data including storage spaces in /QFPNWSSTG.

Note: Remember that Integrated Windows servers must be varied off before starting a save, and be sure to allow enough time for them to shut down properly.

To find out the CL commands that run when you select option 21 (or either of the other options), position your cursor on the option and press the F1 (help) key. The help text provides full details of the objects that are saved with each option. To automate a backup of the complete Windows integration environment, simply include these CL commands in your backup program.

Table 7-3 sets out the individual components of the Windows integration environment and the options on the Save menu that back them up. To use Table 7-3, replace the underlined values with the values that are appropriate for your system. Note that *nwsd-name* is the name of the network server description (NWSD), and *usern* is the name of the *n*th user-defined storage space.

Table 7-3 i5/OS-centric backup using an option from the Save menu

Component	Location	Object type	SAVE Menu options
OS/400 configuration object	QSYS/ <u>nwsd-name</u>	Network Server Description	GO SAVE Option 21, 22, or 23
OS/400 configuration object	QSYS/ <u>nwsd-namexx</u> where xx represents the different types of line descriptions	Line descriptions	GO SAVE Option 21, 22, or 23
Integration code - 5722-WSV - *BASE and option 2	QSYS/QLSVT /QIBM/ProdData/lsvt	Program product	GO SAVE Option 21 or 22
Predefined drive ¹ - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in system, user, or independent user ASP	GO SAVE Option 21 or 23
Predefined drive ¹ - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in system, user, or independent user ASP	GO SAVE Option 21 or 23
User drives ¹	/QFPNWSSTG/ <u>usern</u>	Storage space in system, user, or independent user ASP	GO SAVE Option 21 or 23

¹The integrated Windows server must be varied off for backup of storage spaces to occur.

7.3.7 i5/OS-centric backup using CL commands

Each component of an integrated Windows server can be backed up individually using CL commands from an i5/OS command line. When you perform an i5/OS-centric backup using CL commands, there are three methods you can use:

► **Backing up to tape**

Saving to tape is the most common form of backup. Use the same techniques to save the components of an integrated Windows server as you would use to save any other objects on the system.

You can also use the Backup, Recovery and Media Services (BRMS) licensed program product to provide additional functions and media management capabilities. For more information about BRMS, refer to 7.3.4, “Using BRMS to back up an integrated Windows server” on page 269.

For a detailed description of the CL commands used to save components of the Windows integration environment to tape, refer to “Backing up to tape” on page 272.

► **Backing up to disk**

Saving to disk has the advantage of being very quick, and the data is easily accessible, making for fast restores. Saving and restoring, to and from disk can be used as part of a staged backup strategy.

For a detailed description of the CL commands used to save components of the Windows integration environment to disk, refer to “Backing up to disk” on page 274.

► **Copying to disk**

Copying a storage space to disk is a special case. You can back up a storage space by copying it to disk using the Create Network Server Storage (CRTNWSSTG) command. This operation makes a copy of a storage space under another name.

To back up a storage space by copying it to disk:

- a. Shut down the integrated Windows server.
- b. Use the CRTNWSSTG command to copy the storage space to another of the same size but with a different name. You do not need to unlink the storage space in order to copy it. Here is an example of a command to copy a storage space named SERVER4 to BACKUP4:

```
CRTNWSSTG NWSSTG(BACKUP4) FROMNWSSTG(SERVER4)
```

- c. Restart the integrated Windows server.

Note that you can link the copied storage space to the integrated Windows server as an additional drive in order to retrieve individual files.

For a more detailed description of copying a storage space, refer to Linux in IXS redbook, Implementing Linux on Integrated xSeries Solutions for iSeries, SG24-6379.

Backing up to tape

Table 7-4 on page 273 sets out the CL commands you can use to save the individual components of an integrated Windows server to tape.

To use these commands, replace the underlined values with the values that are appropriate for your system. Note that:

- nwsd-name is the name of the network server description (NWSD).
- iasp-name is the device name for the independent ASP (IASP).
- usern is the name of the *n*th user-defined storage space.

Table 7-4 i5/OS-centric backup to tape using CL commands

Component	Location	Object type	Save command example
OS/400 configuration object	QSYS/ <u>nwsd-name</u>	Network Server Description	SAVCFG DEV(<u>TAP01</u>) ¹
OS/400 configuration object	QSYS/ <u>nwsd-namexx</u> where xx represents the different types of line descriptions	Line descriptions	SAVCFG DEV(<u>TAP01</u>) ¹
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in system ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> '))
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in system ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> '))
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/QASPnn/nwsd-name1.UDFS'</i>)
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/QASPnn/nwsd-name2.UDFS'</i>)
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in independent user ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/iasp-name/nwsd-name1.UDFS'</i>)) ³
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in independent user ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/iasp-name/nwsd-name2.UDFS'</i>)) ³
Integration code - 5722-WSV - *BASE, option 2	QSYS/QLSVT /QIBM/ProdData/lsvt	Program Product	SAVLICPGM LICPGM(5722WSV) DEV(<u>TAP01</u>) ² SAVLICPGM LICPGM(5722WSV) DEV(<u>TAP01</u>) OPTION(2) ²
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in system ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>usern</u> '))
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/QASPnn/usern.UDFS'</i>)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in independent user ASP	SAV DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/iasp-name/usern.UDFS'</i>)) ³

¹SAVCFG saves all configuration objects on the system. You cannot save individual lines or network server descriptions.

²There are two components of the 5722-WSV program product that contain the Windows integration support code:

- *BASE
- Option 2

³For network server storage spaces created in an independent user auxiliary storage pool (IASP), verify that the IASP device is varied on prior to saving the storage space.

Backing up to disk

Table 7-5 sets out the CL commands you can use to save the individual components of an integrated Windows server to disk.

Before you save objects to disk, you must create a save file in a library by using the Create Save File command (CRTSAVF). For example:

```
CRTSAVF FILE(library/save-file)
```

You must use a different save file for each save operation.

To use the commands in this table, replace the underlined values with the values that are appropriate for your system. Note that:

- ▶ nwsd-name is the name of the network server description (NWS D).
- ▶ iasp-name is the device name for the independent ASP (IASP).
- ▶ user n is the name of the n^{th} user-defined storage space.

You can also use the Data compression parameter (DTACPR) on the commands to minimize the disk storage space used. Compressing the data provides better performance and results in a shorter save time. When you save to a save file, you must manually select compression because the command default for the Data Compression parameter (DTACPR) is *DEV, which does not compress a save file.

Table 7-5 i5/OS-centric backup to save files using CL commands

Component	Location	Object type	Save command example
OS/400 configuration object	QSYS/ <u>nwsd-name</u>	Network Server Description	SAVCFG DEV(*SAVF) SAVF(<u>library/save-file</u>) DTACPR(*YES) ¹
OS/400 configuration object	QSYS/ <u>nwsd-namexx</u> where xx represents the different types of line descriptions	Line descriptions	SAVCFG DEV(*SAVF) SAVF(<u>library/save-file</u>) DTACPR(*YES) ¹
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in system ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ')) DTACPR(*YES)
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in system ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ')) DTACPR(*YES)
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in user ASP nn , where nn is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') ('dev/QASP <u>nn</u> / <u>nwsd-name1</u> .UDFS')) DTACPR(*YES)

Component	Location	Object type	Save command example
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/QASPnn/nwsd-name2.UDFS'</i>) DTACPR(*YES)
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in independent user ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/iasp-name/nwsd-name1.UDFS'</i>)) ³ DTACPR(*YES)
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in independent user ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/iasp-name/nwsd-name2.UDFS'</i>)) ³ DTACPR(*YES)
Integration code - 5722-WSV - *BASE, option 2	QSYS/QLSVT /QIBM/ProdData/lsvt	Program Product	SAVLICPGM LICPGM(5722WSV) DEV(*SAVF) SAVF(<u>library</u> / <u>save-file</u>) DTACPR(*YES) ² SAVLICPGM LICPGM(5722WSV) DEV(*SAVF) OPTION(2) SAVF(<u>library</u> / <u>save-file</u>) DTACPR(*YES) ²
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in system ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ')) DTACPR(*YES)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/QASPnn/usern.UDFS'</i>)) DTACPR(*YES)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in independent user ASP	SAV DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/iasp-name/usern.UDFS'</i>)) ³ DTACPR(*YES)

¹SAVCFG saves all configuration objects on the system. You cannot save individual lines or network server descriptions.

²There are two components of the 5722-WSV program product that contain the Windows integration support code:

- *BASE
- Option 2

³For network server storage spaces created in an independent user auxiliary storage pool (IASP), verify that the IASP device is varied on prior to saving the storage space.

7.3.8 i5/OS-centric backup using CL programs

You can automate i5/OS-centric backup on the System i5 by including in your backup program the CL commands that correspond to the save operations that you want to perform. The CL commands that you can use are listed in Table 7-4 on page 273 and Table 7-5 on page 274. Using these commands, you could perform the following types of i5/OS-centric backup unattended:

- System drive, installation drive, and user-defined drives
- The complete Windows environment

We describe both of these scenarios in this section and provide examples of simple CL programs that you can adapt to your own requirements.

To save the complete Windows environment as an automatic process, you can also use option 21 on the i5/OS Save menu, as described in 7.3.6, “i5/OS-centric backup using the Save menu” on page 270.

When you issue a shut down command to an integrated Windows server from System i5 Navigator or an i5/OS command line, the shut down command is passed to the server to enable it to shut itself down cleanly. By default, i5/OS waits 15 minutes (900 seconds) before it performs a forced shutdown of an integrated Windows server, assuming that the server has not already shut itself down. eServer i5/OS V5R3 allows you to change the default wait time to control how long i5/OS waits before forcing a shutdown. You can change the Shutdown Timer as follows:

1. Shut down the server.
2. On an i5/OS command line, enter the Change Network Server Description (CHGNWSD) command and press F4.
3. Scroll down to the Shutdown time-out (SHUTDTIMO) parameter and enter the new default value. Press Enter.
4. Restart the server.

You need to insert a Delay Job (DLYJOB) of at least 900 seconds in your CL program if you use the default time-out, or a DLYJOB value of at least the SHUTDTIMO value in the case of eServer i5/OS V5R3, to allow for the worst case scenario where the server suffers a forced shutdown. Add 30 seconds to the DLYJOB value to allow for the time it takes for i5/OS to force a shutdown. If the delay is not long enough, the CL program fails because the server is not completely shutdown.

Backing up the system and installation drives

Saving the system and installation drives involves backing up the storage spaces that correspond to these drives. You can save these storage spaces to tape or disk. Figure 7-5 on page 277 shows an example of a CL program that could be used to back up the storage spaces that correspond to the system and installation drives to save files on disk. You could customize the CL program shown in Figure 7-5 on page 277 to suit your requirements.

Backing up to a save file is usually faster than backing up to tape. After you save a storage space to disk you can back it up to tape later. You can choose to leave a copy of the storage space on disk to provide a fast method of recovering critical Windows drives.

To use the CL program, replace the underlined values with the values that are appropriate for your system.

You can also use the Data compression parameter (DTACPR) on the commands to save disk storage space. Compressing the data provides better performance and results in a shorter save time. When you save to a save file, you must manually select compression because the command default is *DEV which does not compress a save file.

User storage spaces could also be backed up using the CL program in Figure 7-5 on page 277. All you need to do is add CL commands to create additional save files and save the user storage spaces to these save files.

```

***** Beginning of data *****
0001.00          PGM                                040630
0002.00 /*                                           */ 040630
0003.00 /* THIS PROGRAM BACKS UP THE SYSTEM AND INSTALLATION DRIVES TO SAVE */ 040630
0004.00 /* FILES CALLED SAVE1 AND SAVE2 IN LIBRARY SAVELNX */ 040630
0005.00 /*                                           */ 040630
0006.00 /* VARY OFF THE NETWORK SERVER DESCRIPTION */ 040630
0007.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*OFF) + 040630
0008.00                      ASCVRYOFF(*YES)                                040630
0009.00          MONMSG      MSGID(CPF0000)                                040630
0010.00          DLYJOB      DLY(930)                                    040630
0011.00 /* CREATE THE SAVE FILE LIBRARY */ 040630
0012.00          CRTLIB      LIB(SAVELNX)                                040630
0013.00          MONMSG      MSGID(CPF0000)                                040630
0014.00 /* CREATE THE SAVE FILE FOR THE SYSTEM DRIVE */ 040630
0015.00          CRTSAVF      FILE(SAVELNX/SAVE1)                        040630
0016.00          MONMSG      MSGID(CPF0000)                                040630
0017.00 /* CREATE THE SAVE FILE FOR THE INSTALLATION DRIVE */ 040630
0018.00          CRTSAVF      FILE(SAVELNX/SAVE2)                        040630
0019.00          MONMSG      MSGID(CPF0000)                                040630
0020.00 /* SAVE THE SYSTEM DRIVE */ 040630
0021.00          SAV          DEV('/QSYS.LIB/SAVELNX.LIB/SAVE1.FILE') + 040630
0022.00                      OBJ('/QFPNWSSTG/REDHAT11') DTACPR(*YES)      040630
0023.00          MONMSG      MSGID(CPF0000)                                040630
0024.00 /* SAVE THE INSTALLATION DRIVE */ 040630
0025.00          SAV          DEV('/QSYS.LIB/SAVELNX.LIB/SAVE2.FILE') + 040630
0026.00                      OBJ('/QFPNWSSTG/REDHAT12') DTACPR(*YES)      040630
0027.00          MONMSG      MSGID(CPF0000)                                040630
0028.00 /* VARY ON THE NETWORK SERVER DESCRIPTION */ 040630
0029.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*ON) 040630
0030.00          MONMSG      MSGID(CPF0000)                                040630
0031.00          ENDPGM                                           040630
***** End of data *****

```

Figure 7-5 Example CL program to save the system and installation drives to save files

Backing up the complete Windows integration environment

You can incorporate CL commands from Table 7-4 on page 273 or Table 7-5 on page 274 into a CL program to save all components of the Windows integration environment. You could use this type of backup to restore your Windows environment on another System i5 server after a disaster has rendered your primary system inoperable.

Figure 7-6 on page 278 shows an example of a simple CL program that can be used to perform a disaster recovery backup of the complete Windows integration environment to tape. This program backs up all components of the environment.

To use the CL program, replace the underlined values with the values that are appropriate for your system.

```

***** Beginning of data *****
0001.00          PGM                                040630
0002.00 /*                                           */ 040630
0003.00 /* THIS PROGRAM BACKS UP THE COMPLETE Windows INTEGRATION ENVIRONMENT TO TAPE */ 040630
0004.00 /*                                           */ 040630
0005.00 /* VARY OFF THE NETWORK SERVER DESCRIPTION */ 040630
0006.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*OFF) + 040630
0007.00                      ASCVRYOFF(*YES)                                040630
0008.00          MONMSG      MSGID(CPF0000)                                040630
0009.00          DLYJOB      DLY(930)                                    040630
0010.00 /* SAVE THE COMMUNICATIONS DEFINITIONS */ 040630
0011.00          SAVCFG      DEV(IAP01) ENDOPT(*LEAVE)                    040630
0012.00          MONMSG      MSGID(CPF0000)                                040630
0013.00 /* SAVE THE INTEGRATION CODE */ 040630
0014.00          SAVLICPGM    LICPGM(5722WSV) DEV(IAP01) ENDOPT(*LEAVE) 040630
0015.00          MONMSG      MSGID(CPF0000)                                040630
0016.00          SAVLICPGM    LICPGM(5722WSV) DEV(IAP01) OPTION(2) + 040630
0017.00                      ENDOPT(*LEAVE)                                040630
0018.00          MONMSG      MSGID(CPF0000)                                040630
0019.00 /* SAVE THE SYSTEM DRIVE */ 040630
0020.00          SAV          DEV('/QSYS.LIB/tap01.devd') + 040630
0021.00                      OBJ((' /QFPNWSSTG/redhat11')) ENDOPT(*LEAVE) 040630
0022.00          MONMSG      MSGID(CPF0000)                                040630
0023.00 /* SAVE THE INSTALLATION DRIVE */ 040630
0024.00          SAV          DEV('/QSYS.LIB/tap01.devd') + 040630
0025.00                      OBJ((' /QFPNWSSTG/redhat12')) ENDOPT(*LEAVE) 040630
0026.00          MONMSG      MSGID(CPF0000)                                040630
0027.00 /* SAVE USER DRIVES (ADD AN ENTRY FOR EACH DRIVE) */ 040630
0028.00          SAV          DEV('/QSYS.LIB/tap01.devd') + 040630
0029.00                      OBJ((' /QFPNWSSTG/redhat13')) 040630
0030.00          MONMSG      MSGID(CPF0000)                                040630
0031.00 /* VARY ON THE NETWORK SERVER DESCRIPTION */ 040630
0032.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*ON) 040630
0033.00          MONMSG      MSGID(CPF0000)                                040630
0034.00          ENDPGM                                040630
***** End of data *****

```

Figure 7-6 Example CL program to save the complete Windows integration environment to tape

7.4 i5/OS-centric recovery

We define i5/OS-centric recovery as it relates to running Windows on the Integrated xSeries Server or Integrated xSeries Adapter, as the capability to restore the objects that comprise integrated Windows servers by using i5/OS commands.

Restoring an object or objects from backup is usually a matter of reversing the process you used to save them.

Generally, the best way to perform an i5/OS-centric recovery of the Windows integration environment is by running CL commands. You can also use the i5/OS Restore menu, but it is not as flexible.

In order to restore components of the Windows integration environment, the environment must have been saved using one of the options described in 7.3, “i5/OS-centric backup” on page 265.

7.4.1 Integrated hardware considerations

As part of your recovery plan, you might wish to restore an integrated Windows server and run it on a different Integrated xSeries Server or Integrated xSeries Adapter attached xSeries than the one you saved it on. There are certain restrictions that you must observe when trying to do this. These restrictions result from the fact that the Integrated xSeries Server and xSeries servers are built on Intel-compatible hardware. Windows and Windows Server talk to this hardware using driver programs. Because Intel-compatible hardware varies enormously between manufacturers, and even between different models from the same manufacturer, different drivers are required for the Windows and Windows Server operating systems to talk to the hardware. When you install Windows or Windows Server on Intel-compatible hardware, certain drivers are incorporated into the instance of Windows or Windows Server you are installing, to enable them to talk to the specific hardware you are installing on. Therefore, if you save an instance of Windows or Windows Server that was running on one piece of integrated hardware (Integrated xSeries Server or Integrated xSeries Adapter attached xSeries), you might or might not be able to recover that instance on another Integrated xSeries Server or Integrated xSeries Adapter attached xSeries. It depends on whether or not the drivers are compatible with the target hardware. Note that these issues apply not only to the Integrated xSeries Server and Integrated xSeries Adapter attached xSeries hardware, but to *all* Intel-compatible hardware.

Here are some general guidelines that you should observe if you plan to restore an integrated Windows server instance and run it on a different Integrated xSeries Server or Integrated xSeries Adapter attached xSeries than the one you saved it on:

- ▶ If the integrated Windows server was saved from an Integrated xSeries Server, then it must be restored to run on an Integrated xSeries Server. Running it on an Integrated xSeries Adapter is not supported, and vice versa.
- ▶ If the integrated Windows server was saved from an Integrated xSeries Server model 2892-001 or -002, it can be restored to run on an Integrated xSeries Server model 2892-001 or -002. They are interchangeable.

At the time of writing, the 2892-001 (1.6 GHz) and -002 (2.0 GHz) were the only models of the Integrated xSeries Server supported to run Windows. Later models of the Integrated xSeries Server might or might not be compatible with the 2892-001 and -002. You need to check the Web site:

http://www-03.ibm.com/systems/i/bladecenter/ixs/system_config.html

Note that the 2892-001 and -002 are known by various feature numbers including 4710, 9710, 4810, 2792, 9792, and 2892.

- ▶ If the integrated Windows server was saved from an Integrated xSeries Adapter attached xSeries, then it must be restored to run on an Integrated xSeries Adapter attached xSeries as stated previously. However, if the target xSeries server is in any way different than the source xSeries server (model, model type, or configuration), the Windows target server might or might not start. Here are some guidelines to help you plan for recovery of Windows running on an Integrated xSeries Adapter attached xSeries server:
 - If the source and target xSeries servers are the same model group (for example x235), and exactly the same model type (for example 8671-11X), but with different configurations (for example, different numbers of processors or memory), then the Windows server almost certainly starts, as long as the differences are within Windows's plug and play capability.
 - If the source and target xSeries servers are the same model group (for example x235), but different model types (for example 8671-M1X versus 8671-11X), then the Windows server probably starts, as long as the hardware drivers are compatible.

- If the source and target xSeries servers are different model groups (for example, x365 versus x235), then the Windows server might or might not start, depending whether or not the hardware drivers are compatible. It has been found that some model groups are compatible, but others are not. It depends on the specific source and target model groups.

Unfortunately, there are no hard and fast rules that can be used to predict whether one piece of Intel-compatible hardware is compatible with another. If your recovery planning involves restoring production servers onto backup integrated hardware, you must thoroughly test the recovery environment.

7.4.2 i5/OS-centric recovery roadmap

If you need to recover a complete integrated Windows server, or even a component of one, you need to perform the recovery steps in the following order:

1. “Restoring storage spaces including the system drive”
2. “Restoring configuration objects” on page 281
3. “Relinking storage spaces” on page 281
4. “Recreating TCP/IP interfaces” on page 281

Restoring storage spaces including the system drive

There might be occasions when you want to recover a complete Windows drive, rather than individual files. Because Windows running on the Integrated xSeries Server or Integrated xSeries Adapter is effectively running as a guest operating system under i5/OS, you can restore the drives that comprise the server from backup, rather than rebuilding or reinstalling the server.

The system drive is unique because it contains the Windows kernel and operating system. It is quite possible that this drive can become corrupted because of a faulty application or driver. If you have an unrecoverable system failure that does not allow you to start the server, you can simply restore a copy of the system drive from backup and restart the server. You can think of this capability as a “hot spare system drive”.

If you are restoring your Windows integration environment to a backup System i5 system, there might not be any storage spaces currently on the system. That is, none appear when you use the WRKNWSSTG command. In this case, you must create the /QFPNWSSTG directory before you can restore any storage spaces. To create the /QFPNWSSTG directory, you need to create a dummy storage space as follows:

1. On an i5/OS command line, type CRTNWSSTG and press F4.
2. Provide a name for the storage space.
3. Use the minimal size allowed (say 2 MB) and specify the appropriate disk pool (ASP).
4. Press Enter to create the storage space. i5/OS creates the /QFPNWSSTG directory, and the dummy storage space in it.

To recover a Windows drive (including the system drive) from backup, follow these steps:

1. If you are recovering a Windows drive to an existing server, shut down the server through System i5 Navigator or by using the Vary Configuration command (VRYCFG).
2. If you are replacing a drive, unlink the storage space you want to replace through System i5 Navigator or by using the Remove Server Storage Link (RMVNWSSTGL) command. You can also unlink it from the Work with Network Server Storage Spaces (WRKNWSSTG) menu.

Note: Do not compress the link sequence; otherwise, you will not be able to relink the restored drive in the correct sequence.

3. (Optional) Delete the unlinked storage space through System i5 Navigator or by using the Delete NWS Storage Space command (DLTNWSSTG). You can also delete it from the WRKNWSSTG menu.
4. Use the Restore Object command (RST) to restore the storage space you want to recover from backup, as shown in Table 7-6 on page 284 or Table 7-7 on page 285.

Note that you might need to restore other Windows drives to maintain synchronization between Windows system files, applications, and data.

After you restore a Windows drive, you should restore any incremental backups that you took since the drive was saved. You would need to do this using a file level backup that you created using a Windows backup utility or application.

Restoring configuration objects

In a disaster recovery situation, you might need to restore all the configuration objects for your integrated Windows server, which include the network server description (NWSD) and line descriptions. Note that the controller and device descriptions that are associated with a line description are automatically recreated the first time you start the server. To restore the NWSD and associated line descriptions, refer to the examples shown in Table 7-6 on page 284 and Table 7-7 on page 285.

After all communications objects have been restored, you must redefine your TCP/IP interfaces. This is described in “Recreating TCP/IP interfaces” on page 281.

Relinking storage spaces

The Save Configuration command (SAVCFG) saves the objects associated with an NWSD and the current static (*FIXED) and dynamic (*DYNAMIC) storage space links. Storage spaces are automatically relinked to the restored NWSD providing you restore the storage spaces *before* the NWSD and providing the storage spaces were saved in a linked condition.

If you are recovering an integrated Windows server, you must relink storage spaces in the original sequence. In this case, you must specify a specific drive sequence number (for example, 3 for the system drive, 4 for the installation drive, and so on), not *CALC. Otherwise, you might not link the drives in the correct order and the server will either not start or produce errors. Note that the system and installation drives must always be linked statically as sequence numbers 3 and 4 respectively.

You can link storage spaces through System i5 Navigator, by using the WRKNWSSTG command (option 10) or by using the Add Server Storage Link command (ADDNWSSTGL):

```
ADDNWSSTGL NWSSTG(storage-space) NWSD(NWSD-name) DRVSEQNBR(n)
```

Recreating TCP/IP interfaces

After you restore an integrated Windows server, verify that the associated Point-to-Point Virtual Ethernet LAN TCP/IP interfaces are configured. If you are restoring components of an integrated Windows server on the same System i5 system or partition from which they were saved, then the original TCP/IP interfaces should still be there, unless they have been deleted. If you are restoring components of an integrated Windows server on a different System i5 system or partition from which they were saved, then new TCP/IP interfaces need to be created.

Restriction: At the time of writing this book, no Virtual Ethernet LAN capability is available for integrated Windows servers, but the Point-to-Point Virtual Ethernet LAN TCP/IP interfaces must exist on i5/OS and Windows in order for the server to function. IBM intends to provide a functional Point-to-Point Virtual Ethernet LAN connection at a later date.

To determine if a TCP/IP interface needs to be created, run the `Configure TCP/IP` command (`CFGTCP`), then select option 1. You should see an interface with the same name as your network server description with PP appended, and an IP address of the form 192.168.x.y. If you cannot find this interface, you must create it. You can determine the IP address of the interface by running the `Work with Network Server Descriptions` (`WRKNWSD`) command and entering 5 (Display) to display the network server's configuration. Press Enter until you see a screen that contains the TCP/IP port configuration. Note that the IP address shown actually corresponds to the Windows side of the Point-to-Point Virtual Ethernet LAN. To calculate the IP address of the i5/OS side of the Point-to-Point Virtual Ethernet LAN, simply subtract 1 from the last byte of the address shown against *VRTETHPTP. For example, if the IP address of the *VRTETHPTP port is 192.168.3.2, create a TCP/IP interface for the Point-to-Point Virtual Ethernet LAN line description (ending in PP) using the address 192.168.3.1.

Important: Ensure that the Point-to-Point Virtual Ethernet LAN TCP/IP interfaces are on a unique subnet. Otherwise, you can encounter errors in starting and shutting down the integrated Windows server.

7.4.3 i5/OS-centric recovery methods

There are four basic methods for performing an i5/OS-centric recovery of the Windows integration environment:

- **Using the i5/OS Restore menu**

When you perform a disaster recovery restore of the Windows integration environment from tape, you can use options on the i5/OS Restore menu (GO RESTORE). However, you would not normally recover an integrated Windows server using the Restore menu because you cannot selectively restore individual components. You might overwrite data that you do not want changed. For this reason, we recommend that you use the CL commands described in Table 7-6 on page 284 and Table 7-7 on page 285 to restore components of an integrated Windows server. Therefore, we do not discuss this option further.

- **Using CL commands**

Refer to 7.4.4, "i5/OS-centric recovery using CL commands" on page 283.

- **Using CL programs**

Refer to 7.4.5, "i5/OS-centric recovery using CL programs" on page 286.

- **Using file level recovery via storage spaces**

Refer to 7.4.6, "File level recovery via storage spaces" on page 289.

Note: You cannot perform an i5/OS-centric recovery through System i5 Navigator.

7.4.4 i5/OS-centric recovery using CL commands

When you perform an i5/OS-centric recovery using CL commands, select one of the following methods:

► Restoring from tape

Restoring from tape is the most common form of restore. Use the same techniques to restore the components of an integrated Windows server as you would use to restore any other objects on the system.

You can also use the Backup Recovery and Media Services (BRMS) licensed program product to provide additional functions and media management capabilities. For more information about BRMS, refer to 7.3.4, “Using BRMS to back up an integrated Windows server” on page 269.

For a detailed description of the CL commands used to restore components of the Windows integration environment from tape, see “Restoring from tape” on page 283.

► Restoring from disk

Restoring from disk has the advantage of being very fast, and the data is easily accessible. Saving and restoring, to and from disk, can be used as part of a staged backup strategy.

For a detailed description of the CL commands used to restore components of the Windows integration environment from disk, refer to “Restoring from disk” on page 285.

► Copying from disk

Copying a storage space from disk is a special case. You can restore a storage space by copying it from a backup copy on disk using the Create Network Server Storage command (CRTNWSSTG). This operation makes a copy of a storage space under another name.

To restore a storage space from a copy on disk:

- a. Shut down the integrated Windows server.
- b. Unlink the storage space you are replacing from the network server description. Do not compress the link sequence because this can prevent you from linking the new storage space in the original sequence.
- c. Delete the storage space you are replacing.
- d. (Optional) Use the CRTNWSSTG command to copy the backup storage space to a new storage space of the same size but with the name of the storage space you are replacing. This step is optional because the name does not matter. You could simply link the backup storage space directly to the NWSD. However, you might wish to perform the copy to keep the new storage space name consistent with the original name. Note that you cannot rename a storage space.

Here is an example of a command to copy a storage space named BACKUP4 to SERVER4:

```
CRTNWSSTG NWSSTG(SERVER4) FROMNWSSTG(BACKUP4)
```

- e. Relink the new storage space in the original link position.
- f. Restart the integrated Windows server.

Restoring from tape

Table 7-6 on page 284 sets out the CL commands you can use to restore the individual components of an integrated Windows server from tape.

To use the commands in this table, replace the underlined values with the values that are appropriate for your system. Note that:

- ▶ nwsd-name is the name of the network server description (NWS D).
- ▶ iasp-name is the device name for the independent ASP (IASP).
- ▶ usern is the name of the n^{th} user-defined storage space.

Table 7-6 i5/OS-centric recovery from tape using CL commands

Component	Location	Object type	Restore command example
OS/400 configuration object	QSYS/ <u>nwsd-name</u>	Network Server Description	RSTCFG OBJ(<u>nwsd-name</u>) DEV(TAP01)
OS/400 configuration object	QSYS/ <u>nwsd-namexx</u> where xx represents the different types of line descriptions	Line descriptions	RSTCFG OBJ(<u>nwsd-name</u> *) DEV(TAP01) ¹
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in system ASP	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> '))
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in system ASP	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> '))
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/QASPnn/nwsd-name1.UDFS'</i>))
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/QASPnn/nwsd-name2.UDFS'</i>))
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in independent user ASP	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/iasp-name/nwsd-name1.UDFS'</i>))
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in independent user ASP	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/iasp-name/nwsd-name2.UDFS'</i>))
Integration code - 5722-WSV - *BASE, option 2	QSYS/QLSVT /QIBM/ProdData/lsvt	Program Product	RSTLICPGM LICPGM(5722WSV) DEV(TAP01) ¹ RSTLICPGM LICPGM(5722WSV) DEV(TAP01) OPTION(2) ¹
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in system ASP	RST DEV('/QSYS.LIB/TAP01.DEVD') OBJ('/QFPNWSSTG/ <u>usern</u> '))

Component	Location	Object type	Restore command example
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/QASPnn/<u>usern</u>.UDFS'</i>)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in independent user ASP	RST DEV('/QSYS.LIB/ <u>TAP01.DEVD</u> ') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<i>'dev/<u>iasp-name</u>/<u>usern</u>.UDFS'</i>)

¹This command also restores the network server description.

²There are two components of the 5722-WSV program product that contain the Windows integration support code:

- *BASE
- Option 2

Restoring from disk

Table 7-7 sets out the CL commands to use for restoring the individual components of an integrated Windows server from disk.

To use the commands in this table, replace the underlined values with the values that are appropriate for your system. Note that:

- ▶ *nwsd-name* is the name of the network server description (NWS D).
- ▶ *iasp-name* is the device name for the independent ASP (IASP).
- ▶ *usern* is the name of the *n*th user-defined storage space.

Table 7-7 i5/OS-centric recovery from save files using CL commands

Component	Location	Object type	Restore command example
OS/400 configuration object	QSYS/ <u>nwsd-name</u>	Network Server Description	RSTCFG OBJ(<u>nwsd-name</u>) DEV(*SAVF) SAVF(<u>library/save-file</u>)
OS/400 configuration object	QSYS/ <u>nwsd-namexx</u> where <i>xx</i> represents the different types of line descriptions	Line descriptions	RSTCFG OBJ(<u>nwsd-name</u> *) DEV(*SAVF) SAVF(<u>library/save-file</u>) ¹
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in system ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> '))
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in system ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> '))
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<i>'dev/QASPnn/<u>nwsd-name1</u>.UDFS'</i>)
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<i>'dev/QASPnn/<u>nwsd-name2</u>.UDFS'</i>)

Component	Location	Object type	Restore command example
Predefined drive - System drive	/QFPNWSSTG/ <u>nwsd-name1</u>	Storage space in independent user ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name1</u> ') (<u>'dev/iasp-name/nwsd-name1.UDFS'</u>)
Predefined drive - Installation drive	/QFPNWSSTG/ <u>nwsd-name2</u>	Storage space in independent user ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>nwsd-name2</u> ') (<u>'dev/iasp-name/nwsd-name2.UDFS'</u>)
Integration code - 5722-WSV - *BASE, option 2	QSYS/QLSVT /QIBM/ProdData/lsvt	Program Product	RSTLICPGM LICPGM(5722WSV) DEV(*SAVF) SAVF(<u>library/save-file</u>) ¹ RSTLICPGM LICPGM(5722WSV) DEV(*SAVF) OPTION(2) SAVF(<u>library/save-file</u>) ¹
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in system ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<u>'dev/iasp-name/usern.UDFS'</u>)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in user ASP <i>nn</i> , where <i>nn</i> is from 2 to 32	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<u>'dev/QASPnn/usern.UDFS'</u>)
User drive	/QFPNWSSTG/ <u>usern</u>	Storage space <i>n</i> in independent user ASP	RST DEV('/QSYS.LIB/ <u>library</u> .LIB/ <u>save-file</u> .FILE') OBJ('/QFPNWSSTG/ <u>usern</u> ') (<u>'dev/iasp-name/usern.UDFS'</u>)

¹This command also restores the network server description.

²There are two components of the 5722-WSV program product that contain the Windows integration support code:

- *BASE
- Option 2

7.4.5 i5/OS-centric recovery using CL programs

You can automate recovery on the System i5 by incorporating into your recovery program the CL commands that correspond to the restore operations that you want to perform. The CL commands that you can use are listed in Table 7-6 on page 284 and Table 7-7 on page 285. Using these commands, you could perform the following types of i5/OS-centric recovery unattended:

- System drive, installation drive, and user drives
- The complete Windows environment

We describe each of these scenarios in this section and provide examples of simple CL programs that you can adapt to your own requirements.

When you issue a shutdown command to an integrated Windows server from System i5 Navigator or an i5/OS command line, the shutdown command is passed to the server to enable it to shut itself down cleanly. By default, i5/OS waits 15 minutes (900 seconds) before it performs a forced shutdown of an integrated Windows server if the server has not already shut itself down. eServer i5/OS V5R3 allows you to change the default wait time of the Vary Configuration (VRRCFG) command to control how long i5/OS waits before forcing a shutdown.

You can change the Shutdown Timer as follows:

1. Shut down the server.
2. On an i5/OS command line, enter the Change Network Server Description (CHGNWSD) command and press F4.
3. Scroll down to the Shutdown time-out (SHUTDTIMO) parameter and enter the new default value. Press Enter.
4. Restart the server.

You need to insert a Delay Job (DLYJOB) of at least 900 seconds if you use the default time-out, or a DLYJOB value of at least the SHUTDTIMO value in the case of eServer i5/OS V5R3, into the CL program to allow for the worst case scenario where the server suffers a forced shutdown. Add 30 seconds to the DLYJOB value to allow for the time it takes for i5/OS to force a shutdown. If the delay is not long enough, the CL program fails because the server is not completely shut down.

Restoring the system drive

Recovering the system drive involves restoring the storage space that corresponds to this drive. This might be useful, for example, if you need to recover the system drive in a hurry. This scenario can be thought of as “system drive hot spare”.

You can restore the system drive from tape or disk. Figure 7-7 on page 288 shows an example of a CL program that you could use to restore the storage space that corresponds to the system drive from a save file on disk. Restoring from a save file is usually faster than restoring from tape, and you do not have to waste time searching for the tape and mounting it. You could customize the CL program shown in Figure 7-7 on page 288 to suit your environment.

The Windows installation drive and user drives could also be restored by using the CL program in Figure 7-7 on page 288. All you need to do is add CL commands to unlink, delete, and restore the storage spaces corresponding to these drives. Note that statically linked (*FIXED) and dynamically linked (*DYNAMIC) storage spaces are automatically relinked, provided they were saved in a linked condition. The system and installation drives are always statically linked with sequence numbers 3 and 4 respectively; you must not try to link them dynamically.

To use the CL program in Figure 7-7 on page 288, replace the underlined values with the values that are appropriate for your system.

```

***** Beginning of data *****
0001.00          PGM                                040630
0002.00 /*                                           */ 040630
0003.00 /* THIS PROGRAM RESTORES THE SYSTEM DRIVE FROM A SAVE FILE */ 040630
0004.00 /*                                           */ 040630
0005.00 /* VARY OFF THE NETWORK SERVER DESCRIPTION */ 040630
0006.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*OFF) + 040630
0007.00                                ASCVRYOFF(*YES) 040630
0008.00          MONMSG      MSGID(CPF0000) 040630
0009.00          DLYJOB      DLY(930) 040630
0010.00 /* UNLINK THE SYSTEM DRIVE STORAGE SPACE */ 040630
0011.00          RMVNWSSSTGL NWSSTG(REDHAT1) NWSD(REDHAT1) RENUMBER(*NO) 040630
0012.00          MONMSG      MSGID(CPF0000) 040630
0013.00 /* DELETE THE SYSTEM DRIVE STORAGE SPACE 040630
0014.00          DLTNWSSTG NWSSTG(REDHAT1) 040630
0015.00          MONMSG      MSGID(CPF0000) 040630
0016.00 /* RESTORE THE SYSTEM DRIVE STORAGE SPACE FROM A SAVE FILE */ 040630
0017.00          RST          DEV('/qsys.lib/save1nx.lib/save1.file') + 040630
0018.00                                OBJ('/qfpnwsstg/redhat1')) 040630
0019.00          MONMSG      MSGID(CPF0000) 040630
0020.00 /* VARY ON THE NETWORK SERVER DESCRIPTION */ 040630
0021.00          VRYCFG      CFGOBJ(REDHAT1) CFGTYPE(*NWS) STATUS(*ON) 040630
0022.00          MONMSG      MSGID(CPF0000) 040630
0023.00          ENDPGM                                040630
***** End of data *****

```

Figure 7-7 Example CL program to restore the system drive from a save file

Restoring the complete Windows environment

You can incorporate CL commands from Table 7-6 on page 284 or Table 7-7 on page 285 into a CL program to recover all components of the Windows integration environment. You could use this type of restore to recover your Windows environment on another System i5 server if a disaster has rendered your primary system inoperable.

Figure 7-8 on page 289 shows an example of a simple CL program that you can use to perform a disaster recovery restore of the complete Windows integration environment from tape. This program restores all components of the environment. Note that you might need to recreate the TCP/IP interface for the Point-to-Point Virtual Ethernet LAN line if it has been deleted, or if you are restoring on another System i5. Remember to start the interface. If required, you could add the CL commands to do this to the example program. Refer to “Recreating TCP/IP interfaces” on page 281. Also, you might not want to restore the Windows integration support software, in which case, you should comment out the lines that restore the integration code.

Additional user drives could also be restored by using the CL program in Figure 7-8 on page 289. All you need to do is add CL commands to restore the storage spaces corresponding to these drives. Note that statically linked (*FIXED) and dynamically linked (*DYNAMIC) storage spaces are automatically relinked when you restore them, provided they were saved in a linked condition and provided that you restore the storage spaces *before* restoring the NWSD. The system and installation drives are always statically linked with sequence numbers 3 and 4 respectively; you must not try to link them dynamically.

If there are no storage spaces currently on the system (none appear when you use the WRKNWSSTG command), you must create the /QFPNWSSTG directory before you can restore any storage spaces. Refer to “Restoring storage spaces including the system drive” on page 280 for a description of how to do this.

To use this CL program, replace the underlined values with the values that are appropriate for your system.

```

***** Beginning of data *****
0001.00          PGM                                          040630
0002.00 /*                                          */ 040630
0003.00 /* THIS PROGRAM RESTORES THE COMPLETE Windows INTEGRATION ENVIRONMENT FROM TAPE */ 040630
0004.00 /*                                          */ 040630
0005.00 /* RESTORE THE INTEGRATION CODE (OPTIONAL) */ 040630
0006.00          RSTLICPGM LICPGM(5722WSV) DEV(IAP01)          040630
0007.00          MONMSG      MSGID(CPF0000)                  040630
0008.00          RSTLICPGM LICPGM(5722WSV) DEV(IAP01) OPTION(2) + 040630
0009.00                      ENDOPT(*LEAVE)                  040630
0010.00          MONMSG      MSGID(CPF0000)                  040630
0011.00 /* RESTORE THE SYSTEM DRIVE */ 040630
0012.00          RST          DEV('/QSYS.LIB/tap01.devd') +    040630
0013.00                      OBJ('/QFPNWSSTG/redhat11'))      040630
0014.00          MONMSG      MSGID(CPF0000)                  040630
0015.00 /* RESTORE THE INSTALLATION DRIVE */ 040630
0016.00          RST          DEV('/QSYS.LIB/tap01.devd') +    040630
0017.00                      OBJ('/QFPNWSSTG/redhat12'))      040630
0018.00          MONMSG      MSGID(CPF0000)                  040630
0019.00 /* RESTORE USER DRIVES (ADD AN ENTRY FOR EACH DRIVE) */ 040630
0020.00          RST          DEV('/QSYS.LIB/tap01.devd') +    040630
0021.00                      OBJ('/QFPNWSSTG/redhat13'))      040630
0022.00          MONMSG      MSGID(CPF0000)                  040630
0023.00 /* RESTORE THE COMMUNICATIONS DEFINITIONS */ 040630
0024.00          RSTCFG      OBJ(REDHAT1*) DEV(IAP01)          040630
0025.00          MONMSG      MSGID(CPF0000)                  040630
0026.00          ENDPGM                                          040630
***** End of data *****

```

Figure 7-8 Example CL program to restore complete Windows integration environment from tape

7.4.6 File level recovery via storage spaces

File level recovery via storage spaces is a technique you can use to restore a single file from a storage space. This technique can enable you to perform all your Windows backup and recovery operations using i5/OS commands, without the need to use Windows utilities.

Take the following scenario. One of your users accidentally deletes a file and asks you to recover it. You back up all your Windows drives (storage spaces) every night, so you have a recent backup of the file that needs to be recovered. If you simply restore the backed up storage space containing the file, it overwrites the currently linked storage space because they both have the same name. Unfortunately, you would lose all the changes to files that have been made since the last backup. Therefore, this would not be a good idea.

Alternatively, you could restore the backed up storage space to a storage space with a different name so that no files are overwritten. You could then dynamically link the restored storage space to the running server and copy the file you need from the newly linked backup storage space to its original location. Therefore, you have been able to recover a single file from a storage space without needing to even shut down the Windows server.

To recover a single file from a storage space, follow these steps:

1. Restore the storage space containing the file you need from tape or disk, renaming the storage space during the restore. Use a command from Table 7-6 on page 284 or Table 7-7 on page 285. In the example shown in Figure 7-9 on page 290, a storage space

named redhat18 is restored from tape and renamed redhat18a. Therefore, it does not overwrite storage space redhat18 from which it was backed up.

Replace the underlined values in Figure 7-9 with values that are appropriate for your setup.

2. Link the restored storage space dynamically if the Windows server is started, or statically if it is shut down.
3. Mount the new drive in Windows.
4. Copy the file you need to recover from the newly linked drive to its original location in the Windows file system.

Restore Object (RST)

Type choices, press Enter.

Device > '/qsys.lib/tap02.devd'

+ for more values

Objects:

Name > '/qfpnwsstg/redhat18'

Include or omit > *INCLUDE *INCLUDE, *OMIT

New object name > '/qfpnwsstg/redhat18a'

+ for more values

Directory subtree *ALL *ALL, *DIR, *NONE, *OBJ, *STG

Output *NONE

Bottom

F3=Exit F4=Prompt F5=Refresh F10=Additional parameters F12=Cancel

F13=How to use this display F24=More keys

Figure 7-9 Restoring and renaming a storage space from tape

7.5 Windows-centric backup and recovery

This section describes the techniques you can use to perform Windows-centric backup and recovery on an Integrated xSeries Server or Integrated xSeries Adapter running Windows.

7.5.1 Overview

Windows-centric backup is inherently file-oriented in nature because Windows backup utilities operate at a file level. It is very difficult to back up an entire Windows drive as a single entity using Windows backup utilities. The only way to save a complete copy of a disk volume using Windows is to buy a third party imaging product. These products require special skills to use, and are not supported for use with integrated Windows servers.

It should be noted that when we discuss Windows file level backup and the applications that can be used to perform this task, we are really talking about Windows flat files, that is, non-database files. In i5/OS, we usually do not draw such a distinction because we use the same backup tools to save both database and non-database files. In Windows, however, you

would usually back up database files using a backup application that is specific to the installed database. Therefore, when we discuss file level backup in this section, it is in the context of flat, or non-database files.

There are other differences between an i5/OS and Windows “backup”. When performing an i5/OS backup, we usually end the partition to restricted state and save objects directly to tape. While the data might be compressed, the objects are written sequentially on the tape and not saved to a single container (archive in Windows terms). When backing up to tape, this is the most logical and efficient way to perform a save. In the Windows world however, a “backup” usually means saving files to a single archive file on disk. The archive file can then either be copied to another Windows server in the network for safekeeping, or dumped to tape. In i5/OS terms, this process is essentially what we have previously described in this chapter as staged backup, where we save a library, for example, to a save file on disk and then optionally back up the save file to tape. In this sense, performing an i5/OS save to a save file is similar to performing a Windows backup to an archive. However, an i5/OS administrator would refer to the process as a staged or temporary backup, whereas a Windows administrator would think of it as a normal backup process.

When we discuss backup, we usually assume that we are backing up to tape. However, we can also back up to disk. Backing up to disk is usually an intermediate step before saving to tape in a staged backup. Ultimately we must back up to tape, or to disk on another system to guard against a complete loss of the primary system. While backup to tape is fairly straightforward, there are applications and techniques that can also be used to save data to disk on remote systems so that files can be quickly recovered in the event of a data loss.

Restriction: At the time of writing, the Windows integration software did not support backing up to DVD-RAM.

Although Windows documentation might talk about “disaster recovery” backup, we need to draw a distinction between the disaster recovery backup capability provided by i5/OS for an integrated Windows server and disaster recovery in the context of a Windows backup application. From the Windows point of view, a disaster recovery backup is a backup of all files on a disk volume or volumes versus backing up selected files. However, Windows backup applications still save at a file level, and you can still restore individual files from a Windows “disaster recovery” backup. In contrast, an i5/OS storage space backup saves a complete image of a Windows drive as a single entity, and you cannot directly restore individual files from it. However, i5/OS storage space backup is very fast, and can provide additional functions not readily available with a Windows “disaster recovery” backup, such as the ability to quickly restore a complete copy of the system drive. In the context of this chapter, we regard all Windows backup operations as file level backups.

Important: When using a Windows backup application, you can use either a native System i5 tape drive or a stand-alone Windows server attached tape drive to save and restore Windows files. Both i5/OS and Windows cannot use a System i5 tape drive concurrently; it must be allocated or “locked” to one operating system or the other.

Windows is able to back up files to a System i5 tape drive because a tape device driver is supplied as part of the Windows integration support software. This driver is copied to the Windows installation drive during installation and incorporated into the Windows instance. It provides Windows with the ability to access a range of System i5 tape drives directly. This capability is referred to as *virtual tape*. Note that the tape drive must be varied off in the hosting i5/OS partition. Tape devices show up under the /dev directory, for example, /dev/st0 or /dev/st1.

Using a Windows application to perform file level backups can be difficult to incorporate into an unattended backup from the i5/OS side. This is because Windows cannot share a tape cartridge formatted for use by i5/OS.

Note: Tape libraries and automatic cartridge loaders (ACLs) on the supported devices list are not supported in random access mode. However, they are supported in manual or sequential access mode. For more information, refer to 7.5.5, “Setting up a System i5 tape drive for use by Windows” on page 293.

There are several good reasons why you might want to use a System i5 tape drive in preference to an Integrated xSeries Server or xSeries attached tape drive. System i5 tape drives tend to be very fast, reliable, robust, and high capacity, and it might be possible to consolidate a number of tape drives in your organization down to one or two System i5 devices. If you have multiple integrated Windows servers in the same i5/OS partition, they can all access the same tape drive (although not at the same time). Therefore, you might only need one System i5 tape drive to back up all your Windows servers.

You can find more information about Windows backup strategies and automating the backup process, along with general information about backup and restore, at this Web site:

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

7.5.2 Windows recovery options

In the case where a Windows server fails to start, there are options that allow you to boot from external media and potentially recover the server. These Windows recovery functions include:

- ▶ Rescue diskette
- ▶ Recovery mode CD-ROM

The Rescue diskette and Recovery mode CD-ROM functions are intended to provide you with a means of recovering a Windows server, which fails to start. While it might be possible to use either of these techniques to recover an integrated Windows server, they are designed for stand-alone servers. Note that you could only use these techniques with an Integrated xSeries Adapter attached xSeries server because the xSeries can boot directly from the diskette and CD-ROM drives. An Integrated xSeries Server card does not have its own diskette or CD-ROM drives and, therefore, cannot boot from the Rescue diskette or Recovery mode CD-ROM.

If you are unable to recover a failed *stand-alone* Windows server then your only other alternative is to rebuild it. In the case of an *integrated Windows server*, you should rarely, if ever, need to either use the Windows recovery options or rebuild the server because you can save a complete image of the server using i5/OS storage space backup. In this case, all you need to do to recover a failed server is to restore a previously saved copy of the server's drives (storage spaces) and then restart it. Note that you might need to also restore volatile files from your file level backup to make sure that the data is up-to-date.

You can effectively use these techniques to eliminate the need for the Rescue diskette and Recovery mode CD-ROM options that are available to recover a stand-alone Windows server.

Important: The ability to quickly and easily recover a failed Windows server is one of the major benefits of the Windows integration support.

7.5.3 Choosing a tape drive for use by your Windows backup application

Windows backup applications can either save to a System i5 tape drive or a tape drive directly controlled by a Windows server somewhere in the network.

► **Backing up to a native System i5 tape drive**

System i5 tape drives can be accessed by backup applications running on the integrated Windows server. The Windows tape device driver is supplied with the System i5 Integration for Windows Server program product (5722-WSV).

Because a System i5 tape drive appears to an integrated Windows server as a directly connected tape drive, you can also save remote integrated or non-integrated Windows servers to the System i5 tape drive across the network using a utility, such as rsync. In other words, the System i5 tape drive behaves exactly as if it were a native Windows tape drive.

To set up and use a System i5 tape drive from an integrated Windows server, proceed to 7.5.5, “Setting up a System i5 tape drive for use by Windows” on page 293.

► **Backing up to a native Windows tape drive**

If you have a Windows-based backup infrastructure already in place, it is likely that you have tape drives attached to stand-alone Windows servers in the network. Therefore, you can save files to a tape drive attached to an integrated Windows server in the same way as you would save files on any other stand-alone Windows server in the network.

7.5.4 Restricting System i5 tape drives that can be used by Windows

You can optionally restrict which tape and optical drives can be allocated to integrated Windows servers in the hosting i5/OS partition. You might want to do this in order to reserve certain drives for use by i5/OS only.

Tape and optical drives that are not to be made available to integrated Windows servers can only be specified after installation has been completed. To restrict devices, follow these steps:

1. Shut down the integrated Windows server.
2. On an i5/OS command line, enter the Change Network Server Description (CHGNWSD) command and press F4.
3. Scroll down to the Restrict device resources (RSTDEVRSC) parameter and list the devices not to be made available. Press Enter.
4. Restart the server.

7.5.5 Setting up a System i5 tape drive for use by Windows

This section describes the tasks you need to perform to set up a System i5 tape drive for use by an integrated Windows server.

The System i5 supports a wide range of tape drives. The Windows integration support software provides SCSI drivers for System i5 tape drives so that Windows can access a range of System i5 tape devices. If you have multiple tape drives on your System i5, each one can be allocated separately to either i5/OS or Windows.

A number of System i5 tape drives have been tested for use with integrated Windows servers, but some models cannot be used. The latest information about which tape devices have been tested for use by Windows running on the Integrated xSeries Adapter or Integrated xSeries Server can be found at the Web site:

http://www.ibm.com/servers/eserver/Systemi5/integratedxseries/Windows/tape_support.html

To allocate a native System i5 tape drive for use by an integrated Windows server, the System i5 tape drive must be logically detached (varied off) from i5/OS. Then it must be logically attached (locked) to the integrated Windows server so that Windows thinks it has a physical tape drive directly attached. When this has been done, the integrated Windows server can use the System i5 tape drive as if it were a directly attached tape drive.

Tip: The Windows integration support software supports the use of Automatic Cartridge Loaders (ACLs) in manual and automatic modes. This means that if the ACL is in automatic mode when a Windows backup application ejects a full tape, the media is unloaded and the ACL automatically loads the next tape. However, the first tape must always be loaded manually by the user.

To enable a System i5 tape drive for use by Windows backup utilities and applications, you need to:

1. Format tape media for use by Windows.

Refer to “Formatting tape media for use by Windows” on page 294.

2. Transfer control of a *tape drive* from *i5/OS* to Windows.

Refer to “Transferring control of a tape drive from i5/OS to Windows” on page 295.

3. Transfer control of a *tape drive* from Windows to *i5/OS*.

Refer to “Transferring control of a tape drive from Windows to i5/OS” on page 297.

Formatting tape media for use by Windows

The tape media formats used by i5/OS and Windows are mutually exclusive; i5/OS uses labels, Windows requires a non-labelled tape. Therefore i5/OS and Windows servers cannot share the same tape media.

Important: All new tapes must be initially formatted using the Initialize Tape (INZTAP) CL command. After the tape has been formatted using INZTAP, additional formatting can be done by Windows, if required.

From an i5/OS command line, enter the Initialize Tape (INZTAP) CL command:

```
INZTAP DEV(TAP01) NEWVOL(*NONE) NEWOWNID(*BLANK) VOL(*MOUNTED) CHECK(*NO)
DENSITY(*DEVTYPE) CODE(*EBCDIC)
```

TAP01 is the name of the tape device. Yours might be different.

DENSITY(*DEVTYPE) gives the best performance, however, if you receive an error with this setting when trying to initialize the tape cartridge, try using DENSITY(*CTGTYPE). If you still receive an error, the tape cartridge is incompatible with your tape drive.

This command produces a non-labelled tape that can be used by Windows backup applications.

Note that if tapes are used that are not of the default density for the drive (consult your drive's documentation), you need to reset the tape density after the System i5 partition has been restarted. Follow these steps:

1. Put a spare tape in the drive. Note that the next step erases all the data on the tape.
2. Issue the following command:

```
INZTAP DEV(tape-device-name) CHECK(*NO) Density(*CTGTYPE).
```

3. You can now switch the blank tape with the tape that you want to use for Windows backups. Backup applications should now work normally. Failure to initialize a blank tape of the correct density can have unanticipated results. If you regularly switch tape densities, you might need to repeat the above steps more often than just after an IPL.

Transferring control of a tape drive from i5/OS to Windows

Before you use a System i5 tape drive from a Windows backup application, you must make it unavailable from the i5/OS side using System i5 Navigator or a CL command, and then lock it on the Windows side through a Windows terminal session.

Note that some tape devices report in under more than one device description. Tape libraries (3570, 358x, 3590, and so on) report in as tape libraries (TAPMLBxx) as well as tape devices (TAPxx), where xx is a sequence number. The Windows integration support software does not support the tape library function. Therefore, if your device has a tape library description, both the tape and tape library devices must be made unavailable (varied off) before locking the device on the Windows server. Note that, although tape libraries are not supported as libraries in Windows, you can use them in sequential mode if the drive supports it.

If you have multiple integrated Windows servers being hosted by the same i5/OS partition, only one server at a time can use a particular System i5 tape drive. If you have multiple logical partitions on your System i5, a tape drive that is owned by one partition cannot be shared by integrated Windows servers that are being hosted by other partitions. Note, however, that it might be possible to logically switch tape drives between i5/OS partitions, depending on the hardware configuration of the System i5.

To transfer control of a tape drive from an i5/OS partition to an integrated Windows server, you must have i5/OS Administrator or Backup Operator authority.

To transfer control of a System i5 tape drive from i5/OS to Windows, follow these steps:

1. Using System i5 Navigator:
 - a. From a System i5 Navigator window, expand the i5/OS partition you are working with.
 - b. Expand **Configuration and Service** → **Hardware** → **Tape Devices**.
 - c. Click **Stand-Alone Devices** and then right-click the tape device you want to transfer control of to Windows. Select **Make Unavailable** as shown in Figure 7-10 on page 296.
 - d. If the tape device is also a tape library, click **Tape Libraries** and then right-click the tape library you want to transfer control of to Windows. Select **Make Unavailable**.

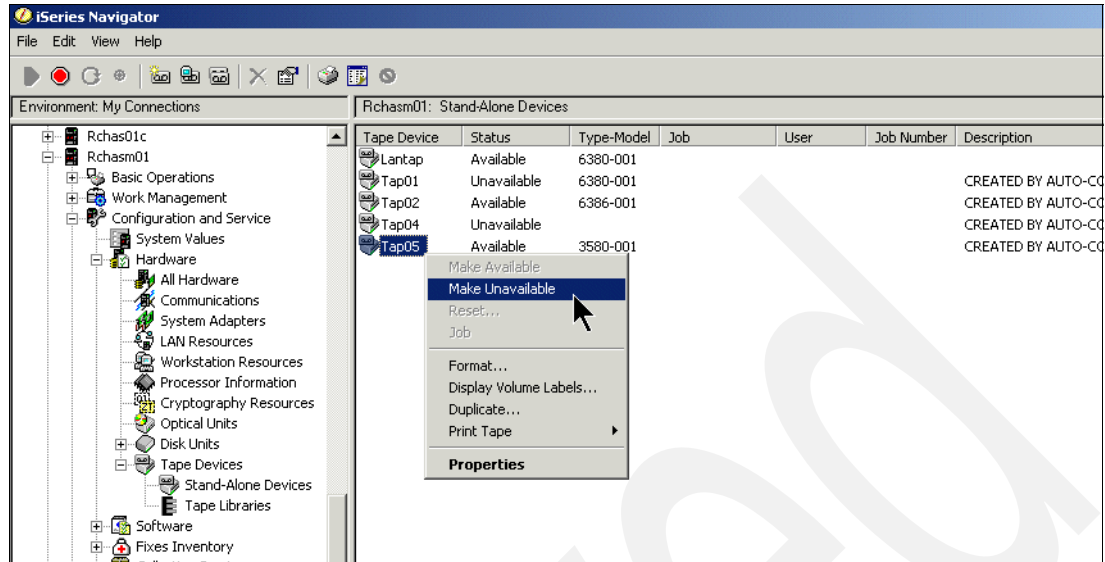


Figure 7-10 Making a tape drive unavailable in System i5 Navigator

2. If you are using an i5/OS command line, follow these steps:

a. On the i5/OS command line, use this command to vary off the tape device:

```
WRKCFGSTS *DEV *TAP
```

On the Work with Configuration Status display, find the tape device you want to transfer control of to Windows. Type 2 next to the device and press Enter.

b. If the tape device is also configured as a tape library, enter the following command:

```
WRKCFGSTS *DEV *TAPMLB
```

On the Work with Configuration Status display, find the tape library corresponding to the tape device you want to transfer control of to Windows. Type 2 next to the tape library and press Enter.

3. To lock the tape device to Windows, start a Windows terminal session and log in as root.

4. At the command prompt, enter the ixsgdev command and press Enter. The syntax is:

```
ixsgdev [-list] | [[-lock | -unlock] device name]
```

For example, to list the tape and optical drives accessible by Windows, enter the following command:

```
ixsgdev -list
```

You should see a display similar to Figure 7-11:

Windows Name	System i5 Name	Type Model	Status

/dev/scd0	OPT01	6321	UNLOCKED
/dev/st0 (/dev/nst0)	TAP01	6380-001	UNLOCKED
/dev/st1 (/dev/nst1)	TAP02	6386-001	UNLOCKED

Figure 7-11 Listing System i5 tape and optical drives in Windows

5. To lock TAP02 to Windows, you would enter the following command:

```
ixsdev -lock TAP02
```

Important: In Windows, commands are case sensitive. Make sure you use the same case as in our examples. For example, use TAP02, not tap02.

You could also use the Windows name:

```
ixsdev -lock /dev/st1 for a rewindable tape device, or
```

```
ixsdev -lock /dev/nst1 for a non-rewindable tape device
```

The `ixsdev -list` command shows the status of TAP02 as **LOCKED**.

Note that you do not need to mount the tape device, because Windows does not see it as a block device. You only need to mount block devices.

6. Insert a tape cartridge that has been formatted for Windows.

After the tape drive has been logically switched to the integrated Windows server, you can use it in the same way as you would use a tape drive directly attached to a stand-alone Windows server. Using a Windows backup application, you can now direct your Windows backups to the System i5 tape drive.

Transferring control of a tape drive from Windows to i5/OS

To hand control of the tape drive back to i5/OS, unlock it on the Windows side and then make it available on the i5/OS side. This procedure is simply the reverse of the process you used to pass control of the tape drive to Windows.

Note that if you shut down the integrated Windows server, or the Windows server fails before you unlock the tape drive, it unlocks automatically. However, it is still in an unavailable state in i5/OS.

To transfer control of a System i5 tape drive back to i5/OS from Windows, follow these steps:

1. To unlock the tape drive from Windows, start a Windows terminal session and log in as root.
2. At the command prompt, enter the `ixsdev` command and press Enter. The syntax is:

```
ixsdev [-list] | [[-lock | -unlock] device name]
```

To unlock TAP02 from Windows, you would enter the following command:

```
ixsdev -unlock TAP02
```

You could also use the Windows name:

```
ixsdev -unlock /dev/st1
```

Important: In Windows, commands are case sensitive. Make sure you use the same case as in our examples. For example, use TAP02, not tap02.

The `ixsdev -list` command shows the status of TAP02 as **UNLOCKED**.

3. If you are using System i5 Navigator:
 - a. From a System i5 Navigator window, expand the i5/OS partition you are working with.
 - b. Expand **Configuration and Service** → **Hardware** → **Tape Devices**.
 - c. Click **Stand-Alone Devices** and then right-click the tape device you want to transfer control of to i5/OS. Select **Make Available** as shown in Figure 7-12 on page 298.

- d. If the tape device is also a tape library, click **Tape Libraries** and then right-click the tape library you want to transfer control of to i5/OS. Select **Make Available**.

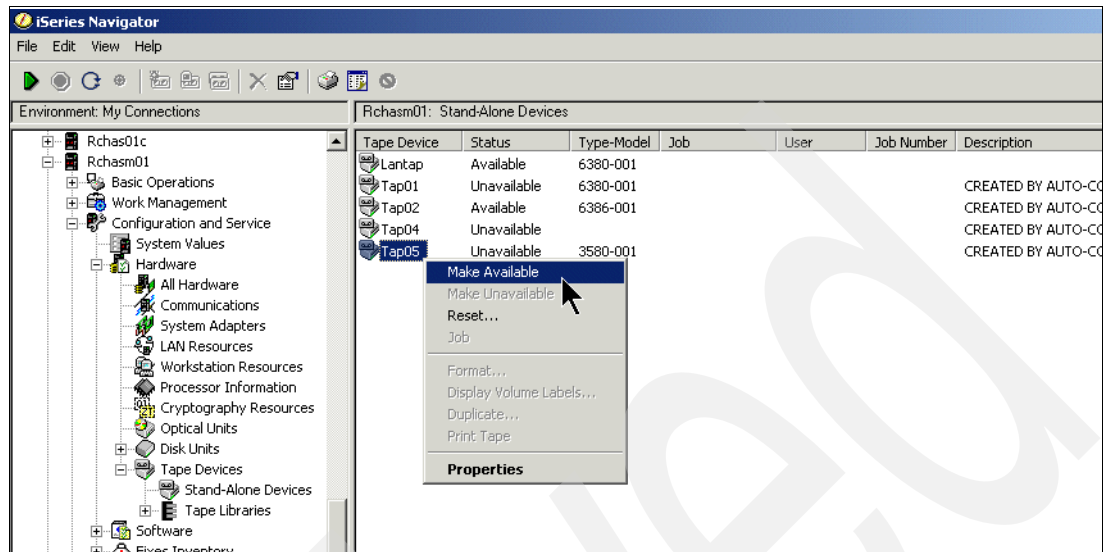


Figure 7-12 Making a tape drive available in System i5 Navigator

4. If you are using an i5/OS command line, follow these steps:
 - a. On an i5/OS command line, use the following command to vary on the tape device:
`WRKCFGSTS *DEV *TAP`
 On the Work with Configuration Status display, find the tape device you want to transfer control of to i5/OS. Type 1 next to the device and press Enter.
 - b. If the tape device is also configured as a tape library, enter the following command:
`WRKCFGSTS *DEV *TAPMLB`
 On the Work with Configuration Status display, find the tape library you want to transfer control of to i5/OS. Type 1 next to the library and press Enter.

You can now use the tape drive from i5/OS.

Windows commercial backup applications

There are other backup applications on the market that offer a wider range of functions than the standard Windows backup utilities. Your main source of information about Windows backup applications such as these is the documentation from the application vendors themselves. These products include:

- ▶ **Tivoli Storage Manager from IBM:**
<http://www.ibm.com/software/tivoli/products/storage-mgr/>
- ▶ **BrightStor ARCserve Backup for Windows from Computer Associates:**
<http://www3.ca.com/Solutions/Product.asp?ID=3370>
- ▶ **Solutions for Enterprise Windows suite of products from Veritas:**
<http://www.veritas.com/Windows/index.jhtml>
- ▶ **Solutions for Windows suite of products from Legato:**
<http://software.emc.com/>

IBM Tivoli Storage Manager

Tivoli Storage Manager is a cross-platform family of products that integrates network backup and archive with storage management and disaster recovery planning functions. It is a sophisticated, powerful, and high performance product that can be used to save files from an integrated Windows server to a tape drive attached either to the System i5 or some other system in the network. Because it is a chargeable product that requires specific skills to use, it is more suited to organizations with a large Windows and Windows Server infrastructure. It could be described as an industrial strength data management environment.

Tivoli Storage Manager implements storage management functions in one single client-server solution covering more than 30 operating platforms. The solution is network-based, which means that the functions are available to the whole network environment. All the functions can be automated to run in a 24 x 7 lights-out environment. Administration costs are minimized by the centralized management of all Tivoli Storage Manager components.

Tivoli Storage Manager is implemented as a client-server application, where the Tivoli Storage Manager server backs up Tivoli Storage Manager clients. All of the storage media used to store Tivoli Storage Manager-managed data is connected to the Tivoli Storage Manager server, known as the storage repository.

There are three major components of Tivoli Storage Manager:

- **Tivoli Storage Manager server**

The Tivoli Storage Manager server is the place where all the Tivoli Storage Manager-managed data is stored. A wide range of systems can act as the server, including System i5. The Tivoli Storage Manager server intelligently stores and manages data from Tivoli Storage Manager clients on disk, tape, or optical devices according to the specified requirements.

- **Tivoli Storage Manager backup client**

The Tivoli Storage Manager backup client is the system to be backed up. It collects the data and sends it to the Tivoli Storage Manager server. The Tivoli Storage Manager client can run on different systems including PCs, workstations, server systems, or the Tivoli Storage Manager server system itself. The clients save their data on the server, including data from databases and applications.

Note: An integrated Windows server can be a Tivoli Storage Manager client.

- **Tivoli Storage Manager administrative client**

For decentralized administration of the Tivoli Storage Manager server and the entire data management environment, Tivoli Storage Manager provides different administration interfaces, also called administrative clients. Depending on the client platform and the Tivoli Storage Manager server version you want to administer, command line interfaces, graphical user interfaces, or web interfaces are available.

7.5.6 Backing up Windows files to the System i5 IFS

You can back up or copy Windows files and archives to a directory in the i5/OS integrated file system (IFS). The Windows files that reside in the IFS could be either direct copies of Windows files or backup archives. This technique provides another option for saving Windows files in addition to, or instead of, backing up files to tape through Windows. Once the Windows files or archives are in the IFS, they can be backed up as part of a normal i5/OS backup procedure.

Windows backup utilities can save files directly to the IFS if an IFS directory can be mapped or mounted to the Windows file system. In this case, the IFS is acting as the file system server and Windows as the file system client. You can establish a client/server relationship between the Windows file system and the IFS using one of the following techniques:

► **Samba and System i5 NetServer**

In this case, you can use the Samba client in Windows to mount an IFS directory shared by System i5 NetServer. For more information, refer to “Samba and System i5 NetServer.”

► **Network File System (NFS)**

NFS is available for both the *System i5* and for Windows. You should be able to set up the System i5 as an NFS server and Windows as an NFS client. You could then mount an *IFS* directory to the Windows file system and then use Windows utilities to back up directly to the IFS. Note that we did not test this scenario. For more information, refer to “NFS” on page 302.

Restriction: When using Windows utilities to back up files directly to the **i5/OS IFS**, using the IFS as the client file system and Windows as the server file system is not supported.

Note that you could also use *File Transfer Protocol* (FTP) to send an archive file directly from Windows to the IFS.

Samba and System i5 NetServer

You can use the capabilities of Samba on the Windows side, in conjunction with System i5 NetServer on the i5/OS side, to save Windows files to a shared directory in the IFS.

System i5 NetServer enables you to share an i5/OS IFS directory and Samba is then able to mount that directory into the Windows file system. This makes System i5 disk storage directly accessible to Windows servers (and clients) running Samba. In addition, when Virtual Ethernet LAN becomes available for Windows integration, you will be able to set up a 1 Gbps internal TCP/IP connection between an integrated Windows server and an i5/OS partition. This connection will provide a fast, reliable, and secure link that enables you to transfer data at high speed between Windows and i5/OS. You will then be able to use the Virtual Ethernet LAN to copy data between Windows running on an Integrated xSeries Adapter or Integrated xSeries Server and an IFS directory shared using System i5 NetServer. Although Virtual Ethernet LAN is not yet available for Windows integration, you can still use an external LAN to connect an integrated Windows server to an i5/OS partition.

There is no real distinction between copying a file and backing it up under Windows in terms of preservation of file attributes. In both cases, you can preserve file attributes such as permission bits and the ownership bit. System i5 NetServer provides the capability to share IFS directories so that you can use a Windows utility to back up or copy the files, together with their attributes, to the IFS. Note that you need to have enough System i5 disk storage available to store the copied files, and you need the correct level of authority to the target IFS directory.

In the following sections, we describe Samba and System i5 NetServer and describe how to set them up so that you can copy files to, or create files in the IFS.

Samba

Samba is an implementation of the Server Message Block (SMB) protocol. It is an open source/free software suite that provides seamless file and print services to SMB clients and comes standard with the supported Red Hat and SUSE distributions. The SMB protocol is used to perform peer-to-peer and client/server-related sharing of network file and print resources.

Microsoft Windows uses the SMB protocol natively. Samba enables you to extend the Windows file and printer sharing environment to platforms that are not running Windows.

You can use a Samba server to authenticate users, and share files, directories, and printers just like Microsoft Windows. Samba can even act as a Primary Domain Controller (PDC) or as a Backup Domain Controller (BDC) in a Windows network. You can use it to run OpenLDAP, and add LDAP function to your Windows Network without additional expense.

You can find detailed information about Samba in the IBM Redpaper titled *Implementing Windows in your Network using Samba*, REDP0023. Search on redp0023 at Web site:

<http://www.ibm.com/redbooks>

This document describes how to install and configure Samba under Windows on an IBM xSeries server. The instructions are the same for Windows on an integrated xSeries server.

You should also refer to the Samba home page to learn more about Samba, its history, and the current and planned implementations:

<http://www.samba.org>

System i5 NetServer

System i5 NetServer comes standard with i5/OS and provides SMB server and client functions similar to Samba. You can use System i5 NetServer to share System i5 integrated file system (IFS) directories to the network. Like Samba, System i5 NetServer is an implementation of the Server Message Block (SMB) server protocol. This is the same protocol that is used by Windows peer-to-peer resource sharing and Windows file servers to share directories to the network.

Although System i5 NetServer is included as standard with i5/OS, you need to install it. Simply use the GO LICPGM command, then option 11 to install i5/OS Host Servers - Option 12 of 5722-SS1. You also need to install the System i5 Navigator component of System i5 Access on a Windows machine in order to create and manage System i5 NetServer shares. For more information about System i5 NetServer, refer to the Web site:

<http://www.ibm.com/servers/eserver/iseri5/netserver/>

The Windows Samba client is also supported by System i5 NetServer. This support allows a Windows client running Samba, which in our case is an integrated Windows server, to connect to System i5 NetServer through the smbmount client utility. The Windows requirement is a kernel version of 2.4.4 or greater and Samba 2.0.7 or greater. For more information about System i5 NetServer support for Windows Samba client, refer to the Web site:

<http://www.ibm.com/servers/eserver/Systemi5/netserver/Windows.htm>

Setting up System i5 NetServer and Samba to use the IFS

To back up or copy Windows files to the i5/OS IFS using System i5 NetServer and Samba, follow these steps:

1. Create a directory in the i5/OS IFS for the Windows files using the MD command. For example:

```
MD DIR('/Windows')
```
2. Using System i5 NetServer, share the directory you have created:
 - a. Start System i5 Navigator from Windows.
 - b. Open your System i5 connection.
 - c. Open **Network**.

- a. Open **Servers**.
- b. Click **TCP/IP**.
- c. Double-click **System i5 NetServer**.
- d. In the System i5 NetServer window, right-click **Shared Objects** and select **New** → **File**.
- e. Click **Browse**.
- f. Click the IFS directory that you created previously. Click **OK**.
- g. Enter a Share name (for example, Windows) and optionally a description.
- h. Change Access to Read/Write. Click **OK**. The shared directory should appear in the right pane of the System i5 NetServer window.

You have now used System i5 NetServer to share an IFS directory on the network.

3. From the Windows console, create a mount point for the System i5 NetServer shared directory:

```
mkdir /mnt/mydata
```

mydata is a mount point for the System i5 NetServer shared directory.

4. From the Windows console, use the Samba client `smbmount` command to mount the share:

```
smbmount //host-name/Windows /mnt/mydata -o  
username=user-name,password=pass-word
```

- host-name is the System i5 TCP/IP host name or IP address.
- user-name is a valid i5/OS profile name.
- pass-word is the password for user-name.

Note that the Samba client is part of the supported Windows distributions. You do not need to set it up.

You can now perform your Windows backup, and specify the shared IFS directory as your target file system. For example:

```
tar -cvpf /mnt/mydata/backup.tar /var
```

Make sure you include the System i5 NetServer shared directory in your i5/OS backup procedure to save the Windows archive as part of your System i5 backup.

NFS

We do not describe Network File System (NFS) in detail because it is a standard Unix and Windows application and is not specific to Windows integration on System i5. You could use NFS to mount an i5/OS directory into the Windows file system. Once mounted, you should be able to back up Windows files directly to the i5/OS IFS for inclusion in an i5/OS backup, although we did not test this scenario.

As with Samba, you must use Windows as the NFS client and i5/OS as the NFS server.

You can find more information about how to set up and use NFS on the System i5 at the Information Center Web site:

<http://publib.boulder.ibm.com/series/>

To navigate to the section on FTP:

1. Select your geographic region.
2. Select your i5/OS release level.

3. Open **Files and file systems**.
4. Open **Integrated file system**.
5. Open **Work with file systems**.
6. Open **Network File System (NFS)**.

FTP

We do not describe File Transfer Protocol (FTP) in detail because it is a standard TCP/IP utility and is not specific to Windows integration on System i5. You could use FTP to send a Windows archive file to the i5/OS IFS for inclusion in an i5/OS backup.

FTP is an application that enables you to *put* files in, or *get* files from a target directory, which can be located in the i5/OS IFS.

You can find more information about how to set up and use FTP on the System i5 at the Information Center Web site:

<http://publib.boulder.ibm.com/series/>

To navigate to the section on FTP:

1. Select your geographic region.
2. Select your i5/OS release level.
3. Open **Networking**.
4. Open **TCP/IP applications**.
5. Open **FTP**.
6. Open **Configure your FTP server**.

Archived

Virtual Ethernet LAN

In this chapter, we discuss the use and installation of Virtual Ethernet LAN as it relates to the implementation of iSCSI network on System i5.

Note that we abbreviate Virtual Ethernet LAN to VE LAN, and point-to-point Virtual Ethernet LAN to PTP VE LAN in this chapter.

8.1 Introduction to Virtual Ethernet LAN (VE LAN)

What is a Virtual Ethernet LAN? As the name implies, Virtual Ethernet LAN is the capability to connect System i5 partitions and integrated servers together using Ethernet protocols but without a physical network of cables, hubs, routers, and switches. Just like a physical Ethernet network, the System i5 Virtual Ethernet LAN uses the TCP/IP protocol, and each node on the LAN has its own IP address. The System i5 Virtual Ethernet LAN enables you to logically connect partitions and integrated servers over an emulated 1 Gb network so that they can exchange information using TCP/IP. The LAN is virtual because it runs across System i5 busses, HSL cables, and iSCSI networks, and is emulated in software running on the System i5.

8.1.1 Basic concepts

Figure 8-1 shows the basic concepts behind VE LAN.

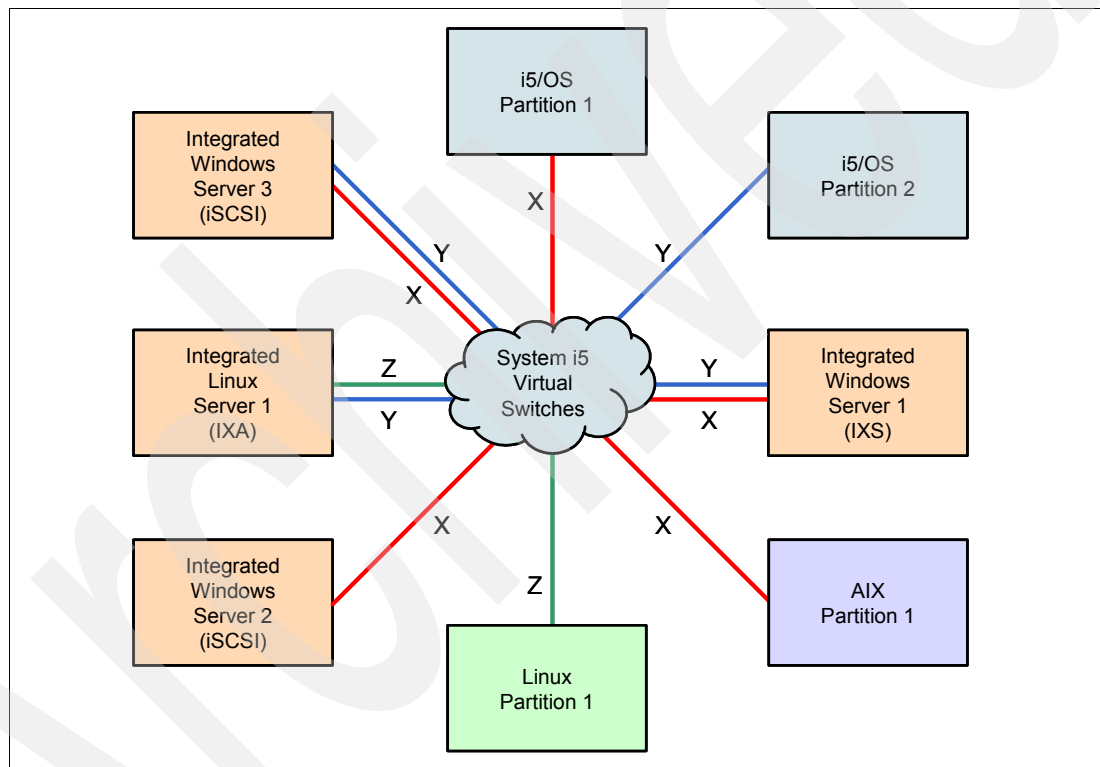


Figure 8-1 VE LAN basic concepts

As you can see in Figure 8-1, System i5 VE LAN can connect:

- ▶ i5/OS partitions
- ▶ AIX partitions
- ▶ Linux partitions
- ▶ Integrated Windows and Linux servers running on IXS and IXA
- ▶ Integrated Windows servers running over an iSCSI network

Not only that, but as Figure 8-1 also shows, these network nodes can be connected on different VE LANs. You can have up to 4095 VE LANs with up to 4094 nodes on each LAN.

8.1.2 Virtual LAN benefits

The Virtual Ethernet is secure because it does not have external connections and is run over the bus. Bus connections make this a very reliable connection method, and the speed of the link is 1Gb, which makes it a very fast connection method. Finally, there is no additional wiring required because the bus connections have already been established.

8.1.3 Virtual LAN limitations

Up to 4094 separate virtual LANs can be defined on each managed system using an HMC. Each i5/OS logical partition can have up to 32 767 Virtual Ethernet adapters defined using an HMC. Virtual Ethernet adapters can use any virtual slot number from slot 2 through slot 65 535 using an HMC.

If you are using the Virtual Partition Manager, you can define up to four separate virtual LANs on each managed system and up to four Virtual Ethernet adapters on each i5/OS logical partition.

Virtual Ethernet adapters created using the Virtual Partition Manager are assigned virtual slot numbers automatically, and you cannot choose which virtual slot number you use. (The slot number used for a Virtual Ethernet adapter displays in i5/OS as the adapter number in the resource details for the CMNxx device.)

i5/OS does not support the IEEE 802.1Q standard. This means that a Virtual Ethernet adapter on an i5/OS logical partition can connect to only one virtual LAN. However, the i5/OS Virtual Ethernet adapter can connect to Virtual Ethernet adapters on logical partitions whose operating systems support the IEEE 802.1Q standard.

8.2 Types of Virtual Ethernet LANs

IP addressing structure for VE LANs over iSCSI

Although not mandatory, we strongly recommend that every point-to-point, non-point-to-point, and iSCSI network is on a different IP subnet.

8.2.1 Point-to-point Virtual Ethernet LAN

The Virtual Ethernet point-to-point connection is the default virtual network connection between the iSeries hosting partition and each integrated Windows server. The point-to-point connection is used primarily for administrative operations, which are part of the integration environment.

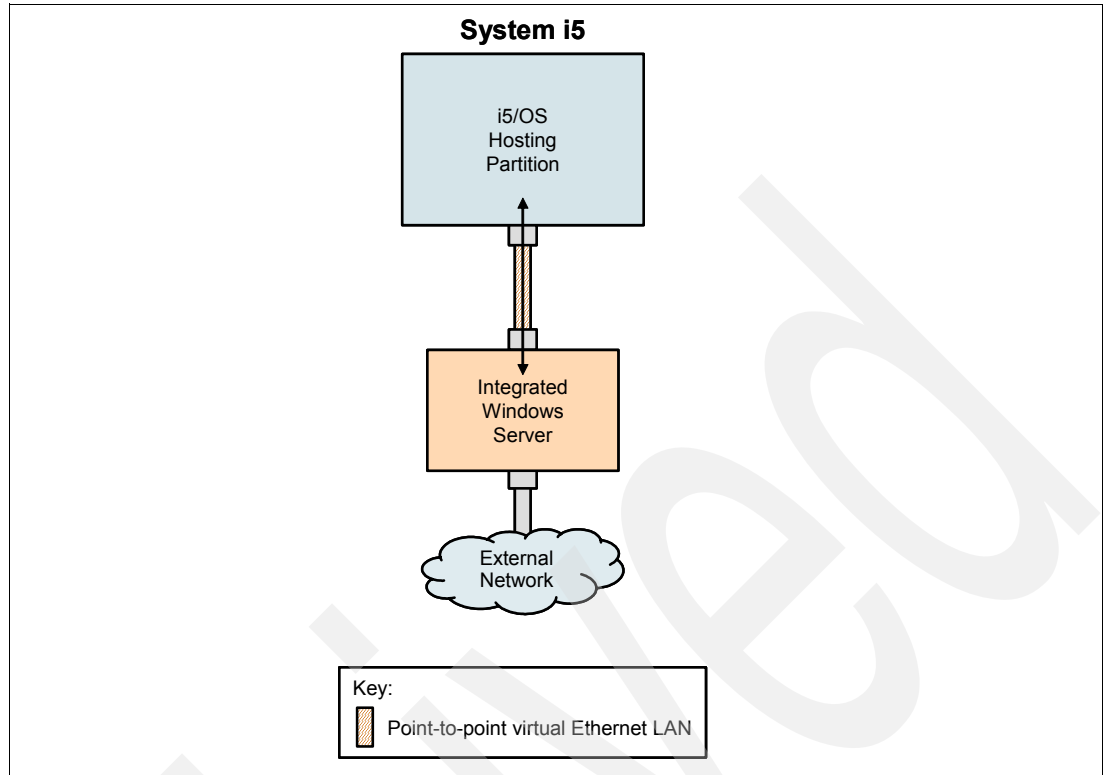


Figure 8-2 Point-to-point VE LAN

i5/OS needs a way to communicate with its integrated Windows servers. This communication takes place over a point-to-point Virtual Ethernet network. When an integrated server is installed, a special virtual network is created between it and a controlling i5/OS partition. This network is called point-to-point because it has only two endpoints, the integrated server and the iSeries, and also because, like a Virtual Ethernet network, it is emulated within the iSeries and no additional physical network adapters or cables are used. In i5/OS, it is configured as an Ethernet line description with Port Number value *VRTETHPTP. When you run the Install Windows server (INSWNTSVR) command, it configures a point-to-point Virtual Ethernet. You might wonder what makes a point-to-point Virtual Ethernet connection different from a Virtual Ethernet network. The answer is that point-to-point Virtual Ethernet is configured differently and can only have two endpoints: the iSeries and an integrated server.

Important: Unless we specifically refer to a point-to-point VE LAN, you can assume we mean a non-point-to-point VE LAN. We refer to a non-point-to-point VE LAN simply as a VE LAN.

IP addressing considerations

Point-to-point Virtual Ethernet only supports the TCP/IP protocol, and, by default, uses restricted IP addresses in private domains, so the addresses are not passed through gateways or routers.

For Integrated xSeries Server (IXS) and Integrated xSeries Adapter (IXA) attached xSeries servers, these addresses take the form of 192.168.xxx.yyy, where (xxx and yyy can be from 1 to 2 digits.) For example, for an IXS that is defined with hardware resource number LIN03, the IP address will be 192.168.3.yyy.

For iSCSI hardware, these addresses take the form of 192.168.xxx.yyy, where xxx ranges from 100 to 254 and results in a unique class C network. In our example, the i5/OS side of the point-to-point network will be given the IP address 192.168.100.1, and the Windows side has 192.168.100.2. As you define multiple line descriptions for the same hardware resource, yyy is incremented. You can allow the INSWNTSVR command to automatically assign these IP addresses or manually configure them to prevent TCP/IP address collisions with other hosts on the system.

8.2.2 Non-point-to-point VE LANs

These are simply referred to as Virtual Ethernet LANs. Like point-to-point Virtual Ethernet, Virtual Ethernet networks are configured through Ethernet line descriptions.

An integrated server is connected to a Virtual Ethernet network when its i5/OS configuration (NWSD) is configured to have an Ethernet line description port number with a value of *VRTETH0 through *VRTETH9. Integrated servers having NWSDs configured with the same port number values are connected to the same Virtual Ethernet network. When installing a new integrated server, the Install Windows server (INSWNTSVR) command can automatically create the required line descriptions and assign them IP addresses.

In Figure 8-2 on page 308, the i5/OS side of the line descriptions is not shown. Unlike when you use Virtual Ethernet, you should configure a TCP/IP address on the i5/OS side of a line description that is used in a Virtual Ethernet network.

Each integrated server can participate in up to four Virtual Ethernet networks simultaneously.

8.3 Inter-partition connections

If you want an integrated server to communicate with other logical partitions, or with integrated servers controlled by other i5/OS partitions, you need to configure one or more inter-partition networks.

8.3.1 Inter-partition networks with the Hardware Management Console

Inter-partition networks are configured differently on iSeries systems with the Hardware Management Console (HMC) than on other systems. In an iSeries HMC system, inter-partition connections exist between partitions or integrated servers using the same VLAN ID. Participating integrated servers do not support VLAN IDs directly. Instead, each participating integrated server needs an Ethernet line description that associates a Virtual Ethernet port value with a virtual adapter having a VLAN ID. The configuration procedure consists of the following steps:

1. Use the Hardware Management Console (HMC) to create a Virtual Ethernet adapter for each partition and each integrated server that will participate in the inter-partition network. See Partitioning with an eServer i5 and Configure Inter-partition Virtual Ethernet networks in the IBM Systems Hardware Information Center for more information. For each virtual adapter that will connect an integrated server or i5/OS partition to the inter-partition network, specify a consistent Port virtual LAN ID and uncheck or clear IEEE 802.1Q compatible adapter.
2. Configure a Virtual Ethernet port and line description, if one has not already been created for the port of interest (0 through 9). Select an associated port name (Cmnxx) for the appropriate 268C resource.

3. Continue with step 2 in all i5/OS partitions that control a participating integrated server.
4. For a partition to fully participate, you will need to appropriately configure the protocols within the partition. In each i5/OS partition, create an Ethernet line description on the appropriate dedicated 268C port resource. Configure an appropriate unique IP address in each partition that will participate in TCP/IP communications.
5. Test to see if the inter-partition network is functioning, for example, by pinging between connected integrated servers and partitions.

Inter-partition networks without the Hardware Management Console

In a system other than an iSeries HMC system, inter-partition connections exist between partitions using the same network number, and integrated servers are connected only if their controlling i5/OS partitions are connected. Network numbers 0-9 are pertinent to integrated servers. For example, if an i5/OS partition is configured for inter-partition communication on networks 1 and 5, then integrated servers controlled by that partition can participate in inter-partition communication on Virtual Ethernet ports 1 and 5. The configuration procedure consists of the following steps:

1. Configure the network number that you want each partition to connect to. Refer to Logical Partition concepts and iSeries Navigator online help information. Keep in mind that integrated servers are connected only if their controlling i5/OS partitions are connected.
2. Configure a Virtual Ethernet port and line description as described if one has not already been created for the port you want to use (0 through 9). Leave the associated port name set to None.
3. Continue in all i5/OS partitions that control a participating integrated server.
4. If you want a partition to fully participate, you need to appropriately configure the protocols within the partition. In each i5/OS partition that you want to participate, use the `WRKHDWRSC *CMN` command to find the name of the appropriate port of hardware type 268C, which was automatically created. See Step 1. Then create an Ethernet line description on the 268C port resource. Configure an appropriate unique IP address in each partition that will participate in TCP/IP communications.
5. Test to see if the inter-partition network is functioning, for example, by pinging between connected integrated servers and partitions.

8.4 VE LAN configuration examples

The following describes some examples of configuring Virtual Ethernet LANs.

8.4.1 Server to non-hosting partition (inter-partition) VE LAN scenario

The System i5 in our example Figure 8-3 on page 311 has been partitioned, creating three separate virtual i5/OS logical partitions inside the iSeries. Three virtual networks are represented in the graphic; two point-to-point Virtual Ethernet networks (in gray and white) and one Virtual Ethernet network (in blue). Each integrated server has a point-to-point Virtual Ethernet network for communicating with its controlling partition.

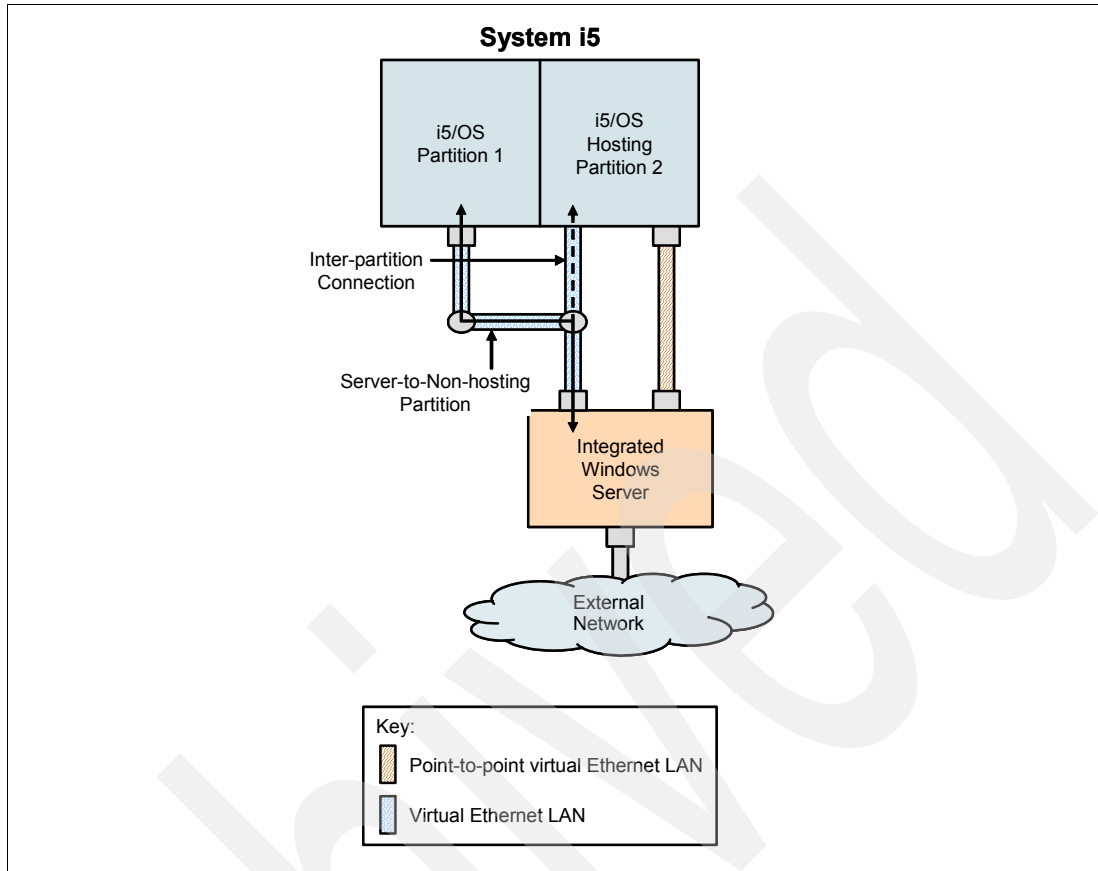


Figure 8-3 Simple inter-partition VE LAN

In this example, the Virtual Ethernet network has three participants: two integrated servers, each controlled by a different i5/OS partition, and a third partition running i5/OS or another operating system. This is called an inter-partition Ethernet network. In servers without a Hardware Management Console (HMC), inter-partition connections exist between partitions using the same network number, and integrated servers are connected only if their controlling i5/OS partitions are connected. Network numbers 0-9 are pertinent to integrated servers.

For example, if an i5/OS partition is configured for inter-partition connections on networks 1 and 5, then integrated servers controlled by that partition can participate in inter-partition communication on ports *VRTETH1 and *VRTETH5. The procedure to do this is in the iSeries Navigator online help. In servers with a Hardware Management Console (HMC), inter-partition connections exist between partitions or integrated servers using the same virtual LAN ID. Participating integrated servers do not support virtual LAN IDs directly. Instead, each participating integrated server needs an Ethernet line description that associates a port value such as *VRTETH1 with a virtual adapter having a virtual LAN ID. You create the virtual adapter using the HMC.

For more information, see the Partitioning with an eServer i5 topic and Configuring a Virtual Ethernet adapter for i5/OS topic in the IBM Systems Hardware Information Center. If you migrate inter-partition Virtual Ethernet from a server without HMCs to a server with an HMC, you need to create Virtual Ethernet adapters using the HMC and additional Ethernet line descriptions to provide appropriate associations. Note that within the same partition, Windows servers can still communicate with each other by simply using the same Virtual Ethernet port number.

8.4.2 Server to server VE LAN scenario - different hosting partitions

In this example (see Figure 8-4), the iSeries has been partitioned, creating three separate virtual i5/OS logical partitions inside the iSeries. Three virtual networks are represented in the graphic; two point-to-point Virtual Ethernet networks (in gray and white) and one Virtual Ethernet network (in blue). Each integrated server has a point-to-point Virtual Ethernet network for communicating with its controlling partition.

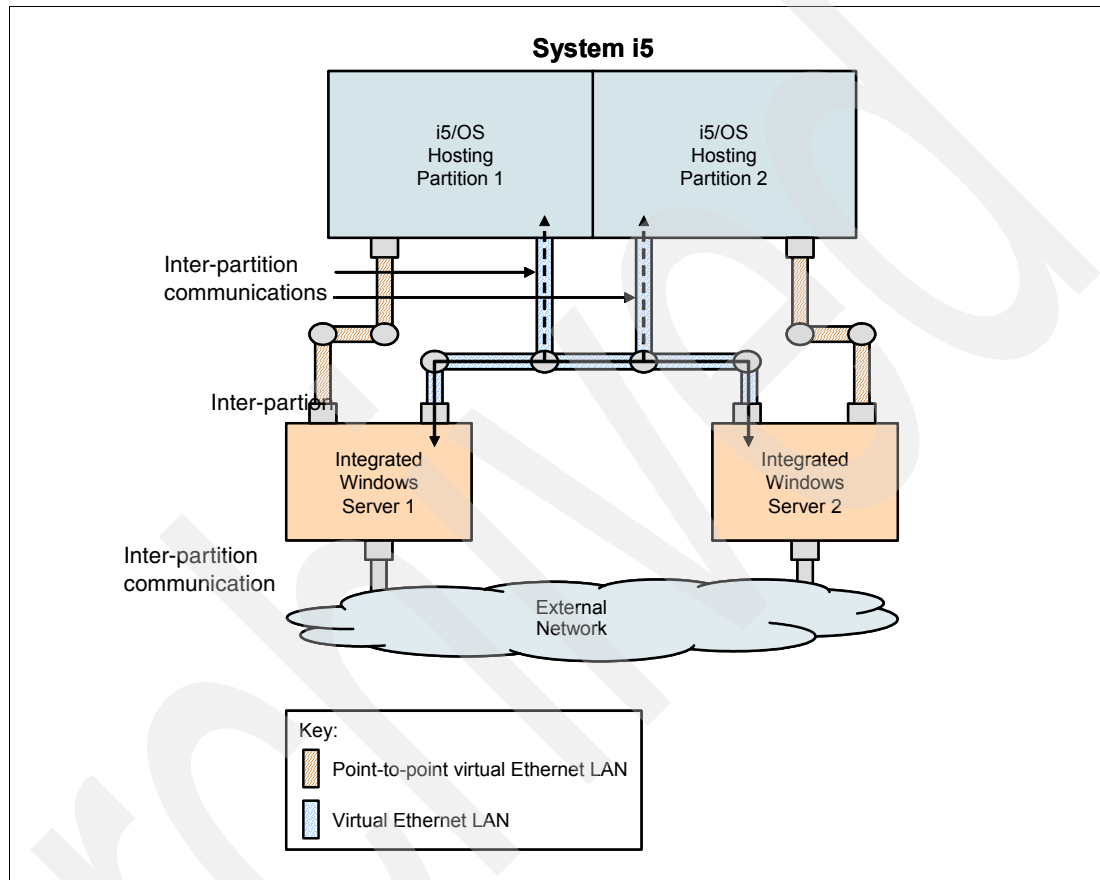


Figure 8-4 Server to server inter-partition VE LAN

In this example, the Virtual Ethernet network has three participants: two integrated servers, each controlled by a different i5/OS partition, and a third partition running i5/OS or another operating system. This is called an inter-partition Ethernet network. In servers without a Hardware Management Console (HMC), inter-partition connections exist between partitions using the same network number, and integrated servers are connected only if their controlling i5/OS partitions are connected. Network numbers 0-9 are pertinent to integrated servers.

For example, if an i5/OS partition is configured for inter-partition connections on networks 1 and 5, then integrated servers controlled by that partition can participate in inter-partition communication on ports *VRTETH1 and *VRTETH5. The procedure to do this is in the iSeries Navigator online help. You can also refer to Logical partition concepts for an overview. In servers with a Hardware Management Console (HMC), inter-partition connections exist between partitions or integrated servers using the same virtual LAN ID. Participating integrated servers do not support virtual LAN IDs directly. Instead, each participating integrated server needs an Ethernet line description that associates a port value such as *VRTETH1 with a virtual adapter having a virtual LAN ID. You create the virtual adapter using

the HMC. For more information, see the Partitioning with an eServer i5 topic and Configuring a Virtual Ethernet adapter for i5/OS in the IBM Systems Hardware Information Center.

If you migrate inter-partition Virtual Ethernet from a server without HMCs to a server with an HMC, you need to create Virtual Ethernet adapters using the HMC and additional Ethernet line descriptions to provide appropriate associations. Note that within the same partition, Windows servers can still communicate with each other by simply using the same Virtual Ethernet port number.

8.4.3 Complex inter-partition VE LAN scenario

Now the iSeries has been partitioned, creating three separate virtual i5/OS logical partitions inside the iSeries. Three virtual networks are represented in the graphic in Figure 8-5 on page 314; two point-to-point Virtual Ethernet networks (in gray and white) and one Virtual Ethernet network (in blue). Each integrated server has a point-to-point Virtual Ethernet network for communicating with its controlling partition.

In this example, the Virtual Ethernet network has three participants: two integrated servers, each controlled by a different i5/OS partition, and a third partition running i5/OS or another operating system. This is called an inter-partition Ethernet network. In servers without a Hardware Management Console (HMC), inter-partition connections exist between partitions using the same network number, and integrated servers are connected only if their controlling i5/OS partitions are connected. Network numbers 0-9 are pertinent to integrated servers. For example, if an i5/OS partition is configured for inter-partition connections on networks 1 and 5, then integrated servers controlled by that partition can participate in inter-partition communication on ports *VRTETH1 and *VRTETH5. The procedure to do this is in the iSeries Navigator online help. You can also refer to Logical partition concepts for an overview. In servers with a Hardware Management Console (HMC), inter-partition connections exist between partitions or integrated servers using the same virtual LAN ID. Participating integrated servers do not support virtual LAN IDs directly. Instead, each participating integrated server needs an Ethernet line description that associates a port value such as *VRTETH1 with a virtual adapter having a virtual LAN ID. You create the virtual adapter using the HMC. For more information, see the Partitioning with an eServer i5 topic and Configuring a Virtual Ethernet adapter for i5/OS in the IBM Systems Hardware Information Center. If you migrate inter-partition Virtual Ethernet from a server without HMCs to a server with an HMC, you need to create Virtual Ethernet adapters using the HMC and additional Ethernet line descriptions to provide appropriate associations. Note that within the same partition, Windows servers can still communicate with each other by simply using the same Virtual Ethernet port number.

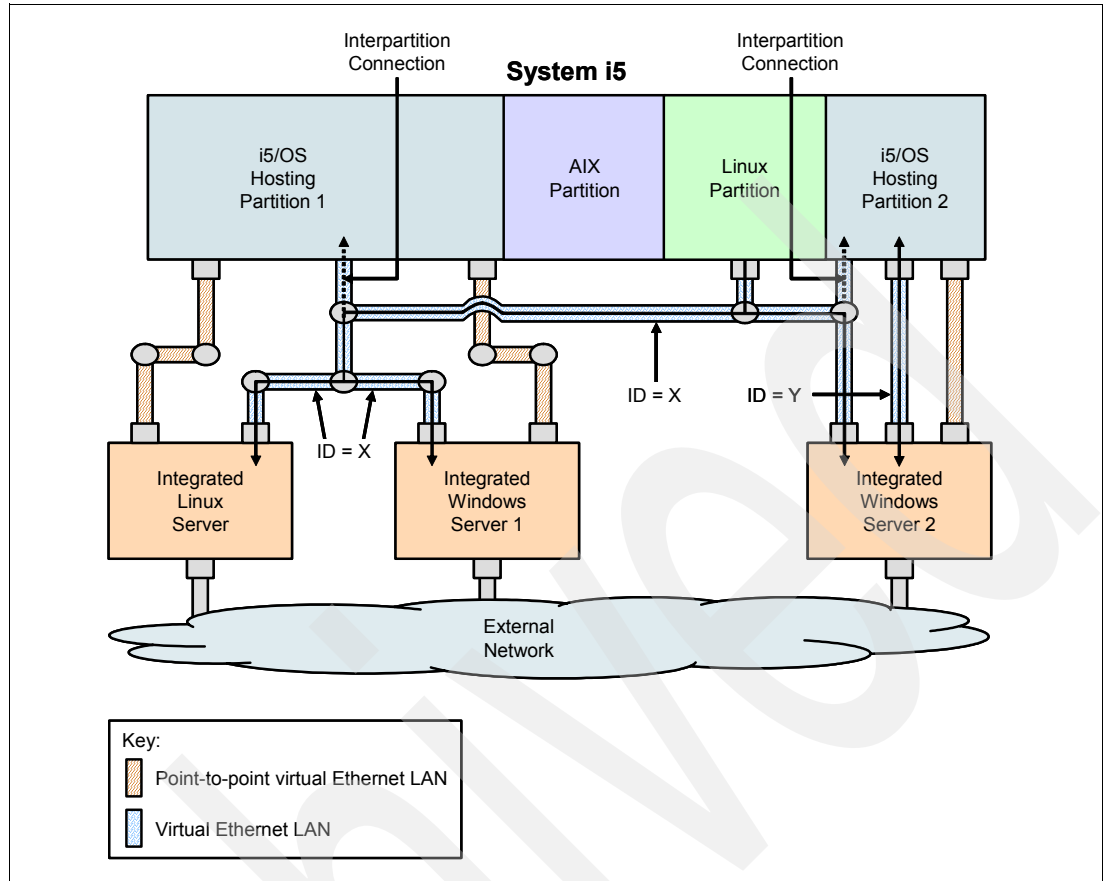


Figure 8-5 Complex inter-partition VE LAN scenario

8.5 Virtual Ethernet LAN IP addressing considerations

The IP address, gateway, and maximum transmission unit (MTU) values for virtual and physical network adapters in the hosted system are managed from the Windows operating system, except for the following cases:

- ▶ The IP address and subnet mask for a new Virtual Ethernet line description can optionally be assigned by the i5/OS Install Windows Server (INSWNTSVR) command. After the server is installed, these values can only be changed from within the Windows operating system.
- ▶ The IP address and subnet mask can be assigned when a Virtual Ethernet line is added to an existing server. After the line description is added, these values can only be changed from within the Windows operating system.
- ▶ Virtual Ethernet point-to-point IP address changes should be configured in both the Windows operating system and i5/OS.
- ▶ The IP address and gateway values for the Windows side of an iSCSI network are always configured and changed from the i5/OS remote system configuration.
- ▶ The IP address, subnet mask, gateway, and MTU values for IXS external LAN adapters can optionally be set in the i5/OS Install Windows Server (INSWNTSVR) command. After

the server is installed, these values can only be changed from within the Windows operating system.

8.6 Virtual Ethernet performance considerations

The iSeries and Windows CPU utilization cost of using the point-to-point connection is similar to the utilization cost of using a hardware network adapter. The connection is high speed, but total bandwidth is always shared with disk, tape, and other operations on IXS and IXA adapters. When you use Internet SCSI (iSCSI), you can separate Virtual Ethernet operations by using another iSCSI HBA channel. A Virtual Ethernet connection between two or more integrated servers uses the iSeries CPU to switch the traffic between servers, even when the iSeries server is not an endpoint of the traffic. For most connections, this utilization is insignificant. But, if you expect high sustained network loads across the Virtual Ethernet connection between integrated servers, you might want to balance the cost of using the Virtual Ethernet internal switch against external network adapters on the integrated servers.

8.7 Setting up VE LAN connections

Setting up VE LAN connections can be broken down into a few distinct tasks:

- ▶ Creating VE LAN resources using the HMC
- ▶ Creating a Virtual Ethernet resource using the HMC
- ▶ Creating a VE LAN connection in the i5/OS hosting partition
- ▶ Creating a simple VE LAN connection for a hosted server

8.7.1 Creating VE LAN resources using the HMC

At V5R4, all VE LAN connections, whether they are in System i5 partitions or hosted servers, require a Virtual Ethernet resource. For example, in Figure 8-6 on page 316, the Virtual Ethernet connection in each one of the System i5 partitions requires a Virtual Ethernet resource, which needs to be created.

Important: Think of a Virtual Ethernet resource as a slot into which you plug a Virtual Ethernet LAN adapter. You need a Virtual Ethernet LAN adapter for every node on the Virtual Ethernet network, either in a partition, or in a hosted server.

You create a Virtual Ethernet resource differently if you have a System i5 with an HMC than you do if your System i5 does not have an HMC. If your System i5 does not have an HMC, refer to 8.7.3, “Setting up a VE LAN connection in an i5/OS partition” on page 321 for a description of how to create VE LAN resources.

8.7.2 Creating a Virtual Ethernet resource using the HMC

To create a Virtual Ethernet resource using an HMC, follow these steps:

1. Select a partition profile as shown in Figure 8-6 on page 316.

Note that you can create Virtual Ethernet resources when the partition is activated or not activated. The difference is that if the partition is activated, you must deactivate it completely (PWRDWN SYS (RESTART *NO)) and then activate it again to create the Virtual Ethernet resources. Here we create the resources while the partition is not activated. You can do it either way.

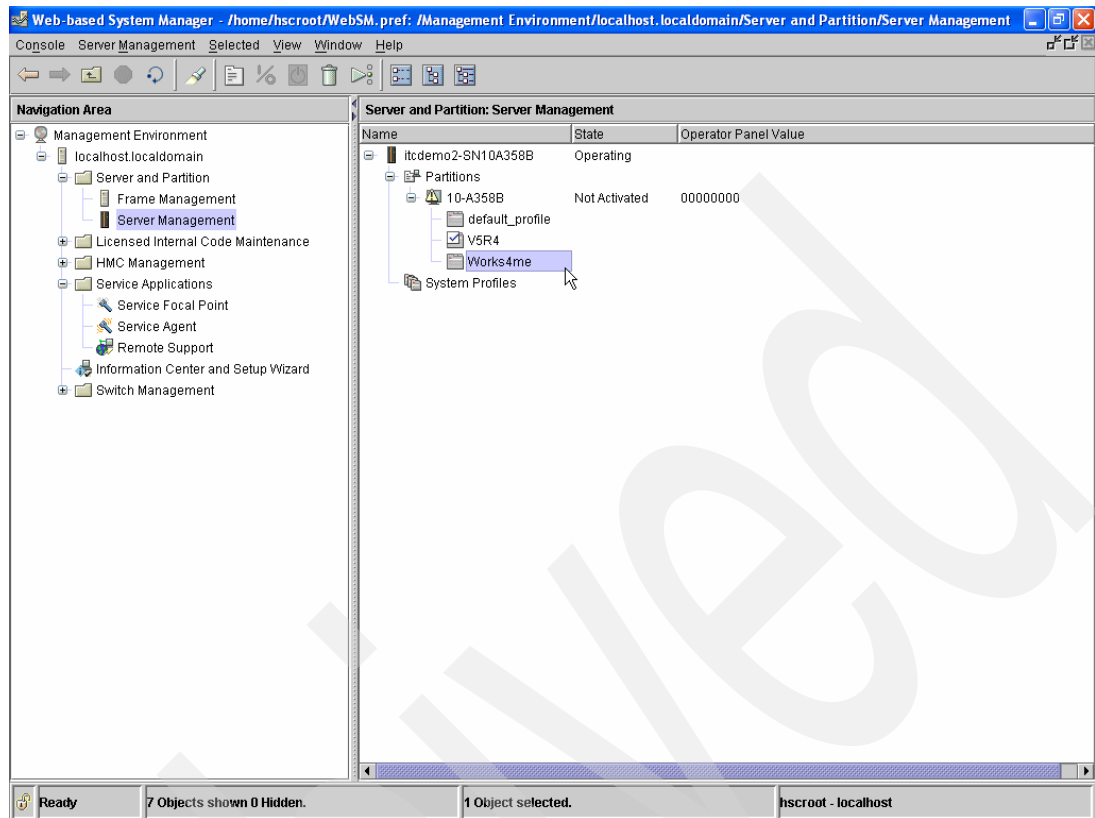


Figure 8-6 Selecting a partition profile

2. Right-click the partition profile and select **Properties**. The window in Figure 8-7 on page 317 appears.

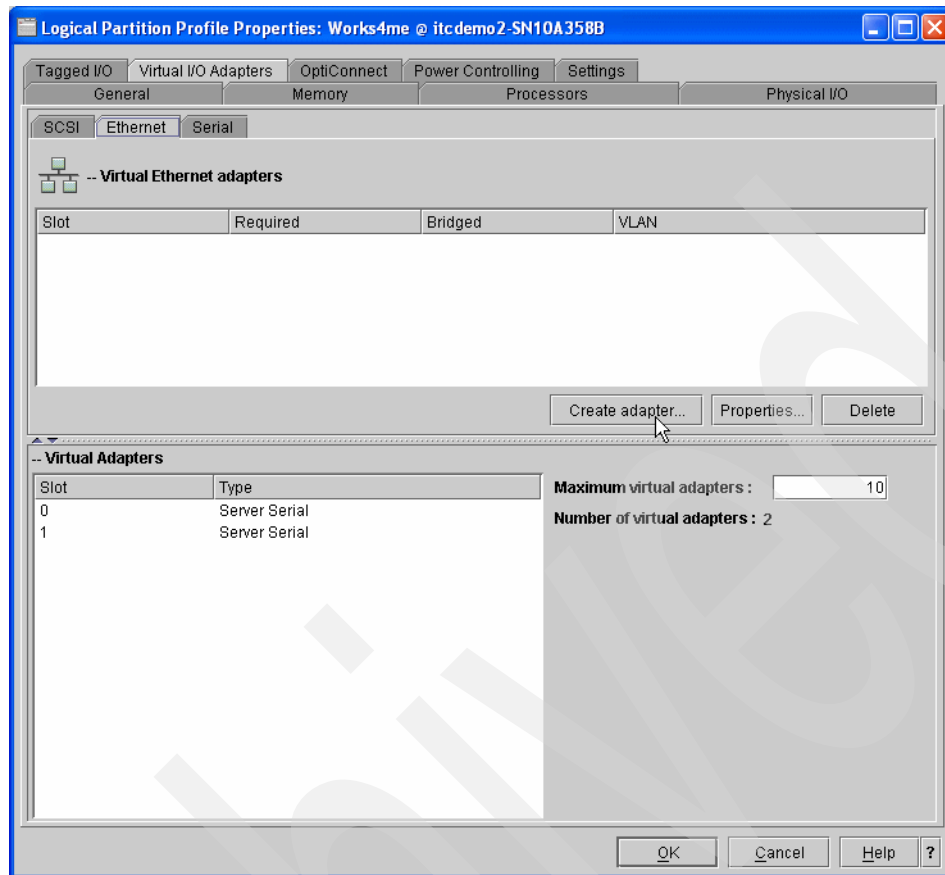


Figure 8-7 Properties window for a partition

3. Select the **Virtual I/O Adapters** tab, then the **Ethernet** tab as shown in Figure 8-8 on page 318.

Note that you might need to increase the Maximum virtual adapters value, depending on the number of Virtual Ethernet resources you want to create. Do not increase this value above a reasonable number, because this uses up Hypervisor memory.

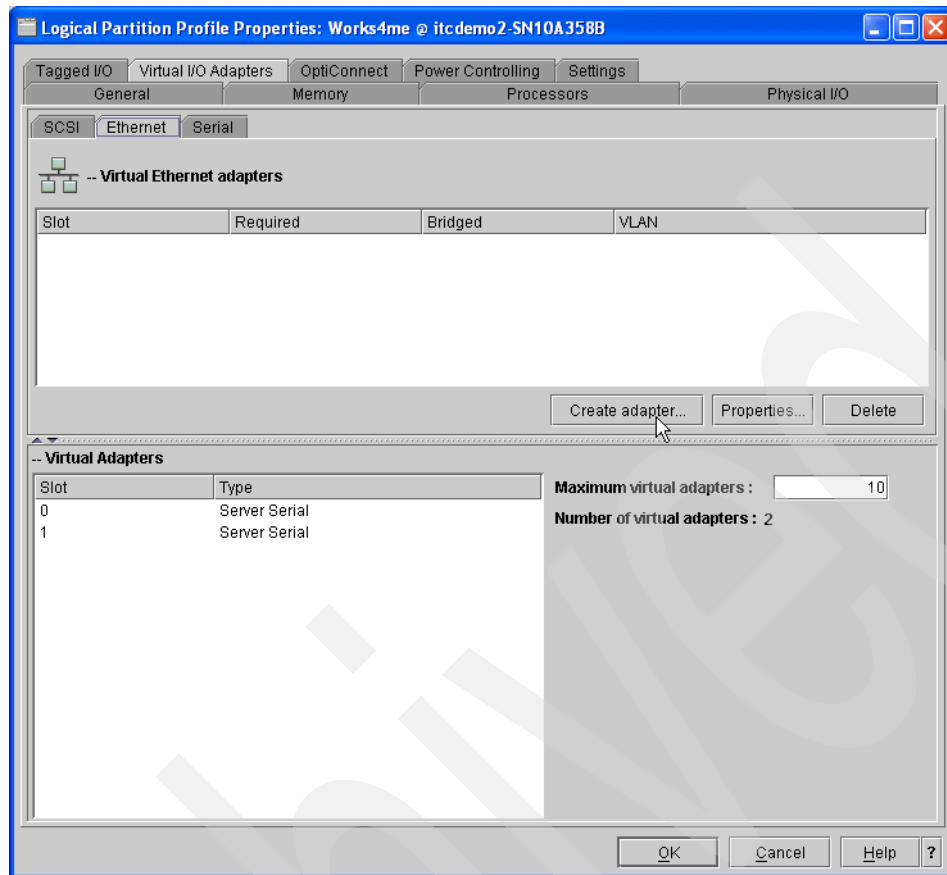


Figure 8-8 Creating Virtual Ethernet resources - 1

- Click **Create adapter**. The window in Figure 8-9 on page 319 appears. The next available slot (2 in this example) and the default Virtual LAN ID (1) are shown in this example.

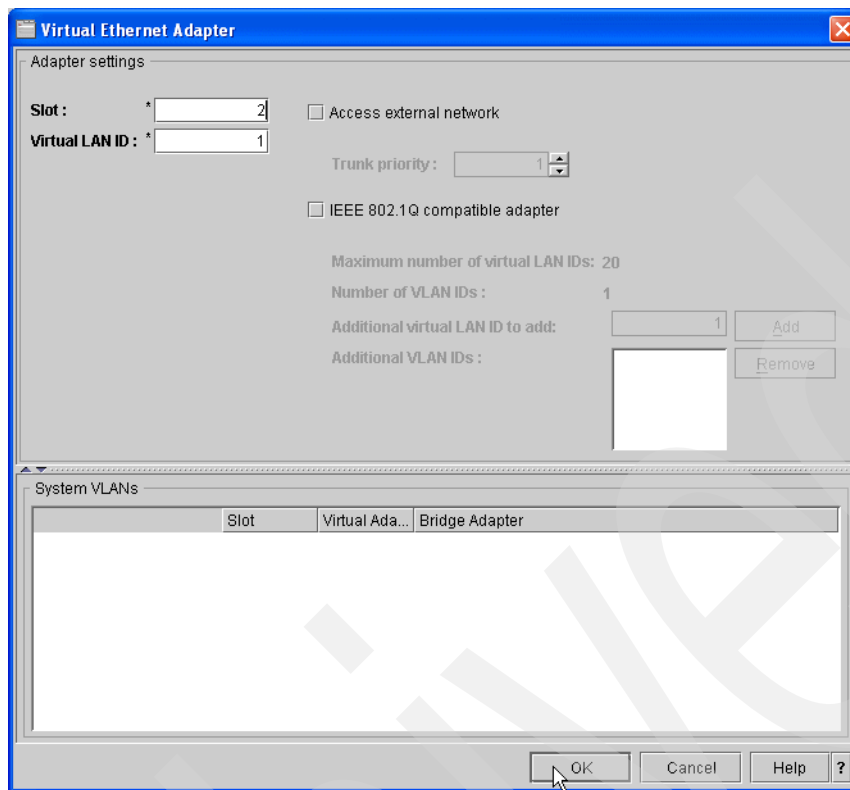


Figure 8-9 Creating Virtual Ethernet resources - 2

Note the following points:

- You need a new slot for each Virtual Ethernet resource you create. It does not matter which slot you use; just take the next available.
- Carefully select the Virtual LAN ID. You can use the default of 1 or select some other value. The important point to note is that all nodes that you want to communicate on the same Virtual Ethernet LAN must use a Virtual Ethernet resource with the *same* Virtual Ethernet LAN ID.

Do not check the Access external network check box, or the IEEE 802.1Q compatible adapter check box. These options are not applicable to System i5.

Click **OK**.

5. You see the new Virtual Ethernet resource that you have just created as shown in Figure 8-10 on page 320. Check the **Required** box to ensure that this Virtual Ethernet adapter resource always starts when the partition is activated.

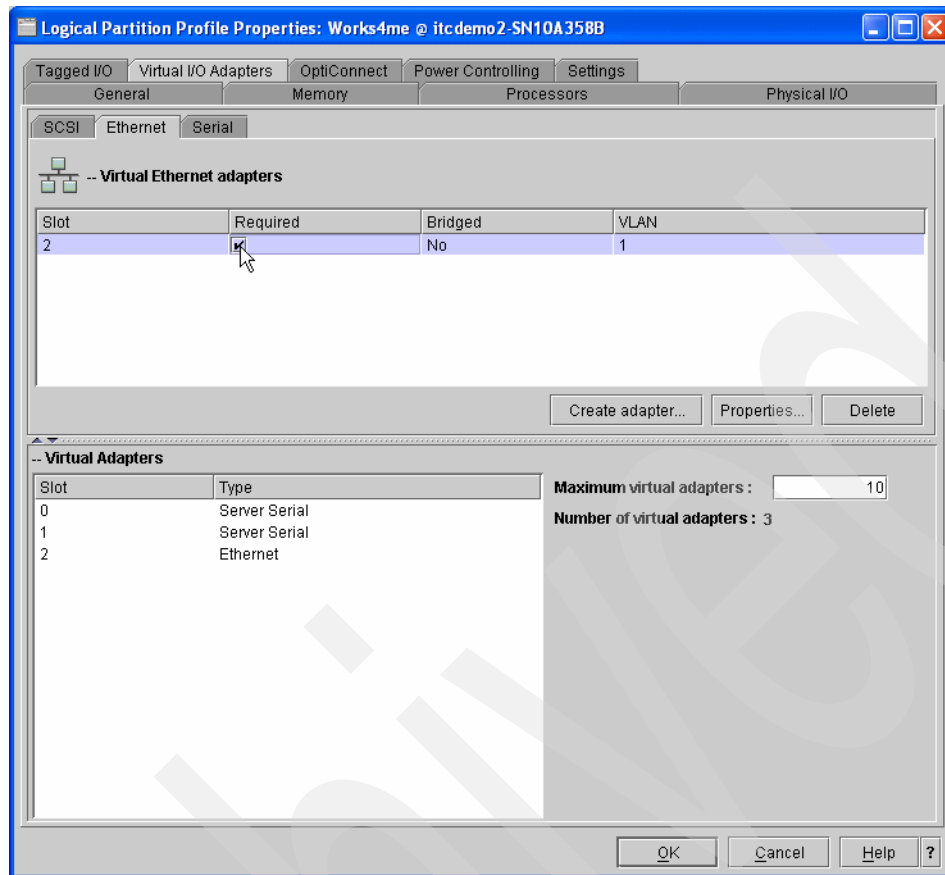


Figure 8-10 Creating Virtual Ethernet resources - 3

6. To create additional Virtual Ethernet resources, simply repeat the same procedure. For example, we want to create two more Virtual Ethernet resources, using the same VLAN ID, so that three nodes can communicate on the same Ethernet network. Therefore, we create a second Virtual Ethernet resource on VLAN 1 using the next available slot (3), and a third Virtual Ethernet resource on VLAN 1 using the next available slot (4). We end up with three Virtual Ethernet resources, all using VLAN 1 as shown in Figure 8-11 on page 321.

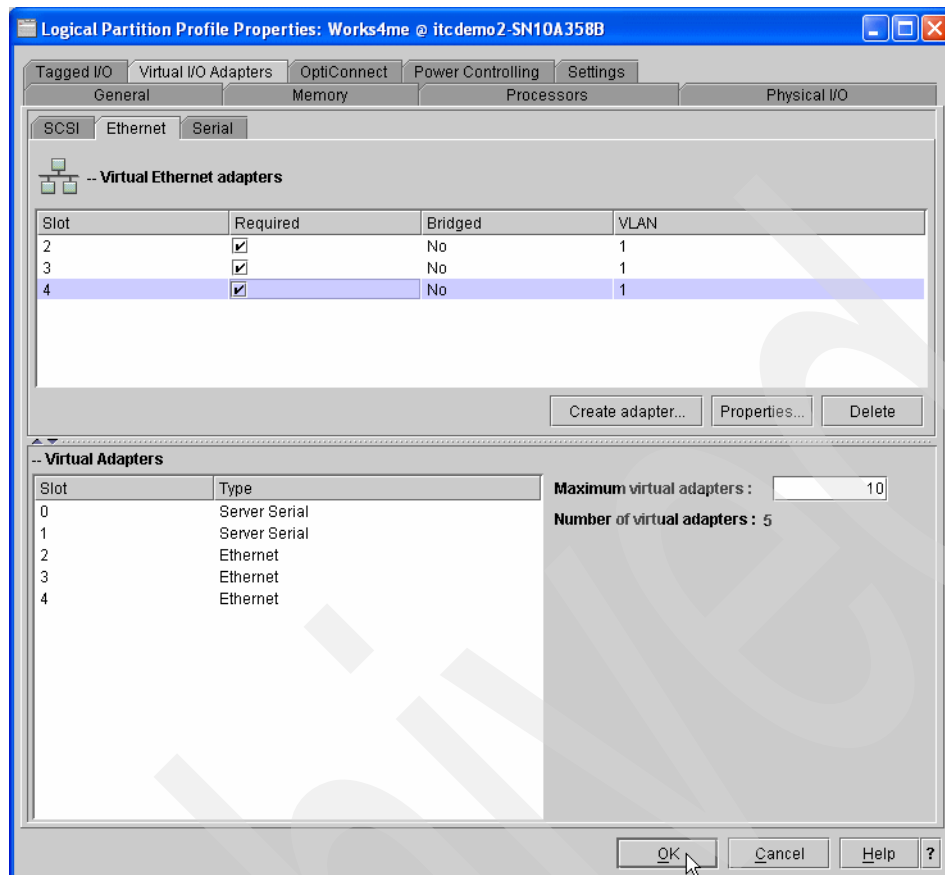


Figure 8-11 Creating Virtual Ethernet resources - 3

We can now allocate these Virtual Ethernet resources to three different nodes so that they can all communicate on the same Virtual Ethernet LAN.

8.7.3 Setting up a VE LAN connection in an i5/OS partition

This section will describe how to configure a Virtual Ethernet network between integrated servers. (Note that if you are installing an integrated server from scratch, the installation command (INSWNTSVR) can configure Virtual Ethernet networks for you.) The procedure consists of the following basic steps:

1. Configure a Virtual Ethernet port and line description for the integrated server. Using iSeries Navigator:
 - a. Expand **Integrated Server Administration** → **Servers**.
 - b. Right-click the integrated server and select **Properties**.
 - c. On the server properties panel, click the **Virtual Ethernet** tab.
 - d. Click **Add** to add a new Virtual Ethernet port.
 - e. On the Virtual Ethernet properties panel, specify the values for the new Virtual Ethernet port:
 - i. Select the Virtual Ethernet port number.
 - ii. Type the IP address that the integrated server will use.
 - iii. Type the subnet mask that the integrated server will use.

- iv. You can leave the default line description name or change it to something else. The default line description name is the NWSD name followed by a v followed by the port number. For example, if adding port 3 to an NWSD named Mynwsd, then the default line description name is Mynwsdv3.
 - v. Leave the associated port set to None.
 - vi. Leave the maximum frame size set to the default 8996.
 - vii. If the server is an iSCSI attached server, select the network server host adapter corresponding to the iSCSI HBA that you want i5/OS to use for this Virtual Ethernet configuration to reach the hosted system.
 - viii. Click **OK** to add the new port to the Virtual Ethernet page on the server properties panel.
 - f. On the server properties panel, click **OK** to save the changes. This updates the NWSD and creates a line description for the new Virtual Ethernet port.
 - g. If you want this integrated server to be connected to more than one Virtual Ethernet network, repeat all of the above steps to create a Virtual Ethernet port and a line description for each network, using different Virtual Ethernet port numbers.
2. Repeat the procedure for all the integrated servers you want to connect to the network, specifying the same Virtual Ethernet port for each one.
 3. Restart the integrated servers. A Virtual Ethernet adapter device driver is automatically installed and set to the Windows TCP/IP address that has been specified for it in the NWSD. However, an IP address entered at the integrated server console overrides the values that are set in the NWSD.
 4. Test to see that the Virtual Ethernet network is functioning, for example by pinging from one server to the IP addresses you specified for the other servers.

8.8 Browsing the Virtual Ethernet LAN topology

Each integrated server has a point-to-point Virtual Ethernet network connection with the iSeries, which allows the iSeries to control the integrated server. Here you can learn how to view or change these connections, although they are automatically configured during installation.

8.8.1 View Ethernet connections from i5/OS

Point-to-point Ethernet connections in i5/OS are composed of a line description and an entry in an integrated server's NWSD.

1. To view the line description, issue the command `WRKCFGSTS *NWS` from the i5/OS character-based interface.
2. Find the cascade of entries corresponding to your integrated server. One of the entries in the Line Description column will have the same name as your NWSD and end with the characters PP. Enter 8 to its left and press Enter.
3. Now you are in the Work with Line Descriptions menu. Enter a 5 (Display) in front of your line description and press Enter to display its information.
4. Press F3 until you return to the base menu.
5. Now issue the command `CFGTCP` and select option **1, Work with TCP/IP interfaces**.

6. One entry in the Line Description column should have the same name as your NWSD and end with the letters PP.
7. Option 5 displays the TCP/IP Interface information, while options 9 and 10 allow you to enable and disable it. Note the Internet address. It is used later.
8. Now, we take a quick look at the entry in the integrated server's NWSD. Issue the command WRKNWSD. Find your integrated server's NWSD and enter 5 (Display) to display it. Press Enter to page through the NWSD attributes.
9. One of the windows is titled Attached lines and displays Port number *VRTETHPTP and the name of the line description that the network is using.
10. Back in the Work with Network Server Descriptions menu, you can use option 2 to change this information.

8.8.2 View Ethernet connections from the integrated Windows server console

This section lists the steps to view the Ethernet connection and optional check steps:

1. At the console of your integrated server, click **Start** → **Settings** → **Control Panel**. Then, select **Network** and **Dial-up Connections**.
2. One of the icons is named Virtual Ethernet point-to-point. Double-click it.
3. Click **Properties** in the dialog box, which appears.
4. Double-click **Internet Protocol (TCP/IP)** in the next dialog box.
5. In this final dialog box, you should see the IP address associated with the integrated server side of the point-to-point Virtual Ethernet connection. It should be the IP address of the i5/OS augmented by one in order to be even instead of odd.

Now, there are steps that enable you to check the IP address found above:

1. Close all of the windows that you opened, click **Start** → **Run**, and enter the command `cmd`. Press Enter. This starts an instance of the Windows command prompt.
2. At the `C:\>` command prompt, which appears, enter the command `ping` followed by the i5/OS IP address, which you remember from the last step. For example, `ping 192.168.3.1`. The command should return:

Reply from x.x.x.x (the standard TCP/IP ping response).

The ping command sends a packet of data to a certain Internet address and times how long it takes to make a round trip.

3. (optional) Return to the i5/OS character-based interface and enter the command `call qcmd`. (This increases the display space so that you can see the results of your commands.) Use the i5/OS command to ping the integrated server. For example, `ping 192.168.3.2`. Congratulations. If all went correctly, we have proved that you have a properly functioning point-to-point Virtual Ethernet network.

Archived

Scaling your iSCSI network

We define the “basic iSCSI configuration” as a one to one (one target HBA to one initiator HBA port) iSCSI connection between a System i i5/OS partition and a hosted xSeries or Blade server. In this chapter, we build on the basic iSCSI configuration you created by working through the instructions on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme.index.html>

We describe how to scale it up to provide additional bandwidth for a specific hosted server connection, or scale it out to connect additional hosted servers.

Important: This chapter assumes that you have already set up a basic iSCSI configuration between an i5/OS hosting partition and a hosted xSeries or Blade server. If you have not, then you need to set up a basic iSCSI configuration by working through the instructions on the Web site above before you can use this chapter.

Before reading this chapter, you should be very familiar with the iSCSI architecture as implemented on System i. If not, we recommend that you work through the documentation on the web, starting at the following Web site:

http://www.ibm.com/systems/i/bladecenter/about_integration.html

This chapter covers the following topics:

► **Overview**

The overview describes the terminology used in the implementation of iSCSI on the System i5 and introduces the concepts of scaling as they apply to a System i5 iSCSI network.

Refer to 9.1, “Overview” on page 327.

► **Increasing the bandwidth of a hosted server connection**

We discuss how you can scale up the bandwidth of the connection between a hosted server and its hosting i5/OS partition.

Refer to 9.2, “Increasing the bandwidth of a hosted server connection” on page 331.

► **Increasing the number of hosted servers**

We discuss how you can scale up the System i iSCSI network by adding additional hosted servers.

Refer to 9.3, “Increasing the number of hosted servers” on page 340.

► **Scenarios for scaling a hosted xSeries or Blade server connection**

We present a number of scenarios which show how a hosted xSeries or Blade server connection can be scaled up in terms of bandwidth and number of servers. We also go through the benefits and considerations of each scenario, and how you configure it.

Refer to:

- 9.4, “iSCSI configuration nomenclature” on page 341
- 9.5, “Scenarios for scaling a hosted xSeries server connection” on page 343
- 9.6, “Scenarios for scaling a hosted Blade server connection” on page 352

► **Path deployment**

We describe the automatic and manual deployment of storage space links and Virtual Ethernet LAN connections to target and initiator HBA ports.

Refer to:

- 9.7, “Introduction to path deployment” on page 363
- 9.8, “Automatic path deployment” on page 369
- 9.9, “Manual path deployment - target side” on page 377
- 9.10, “Manual path deployment - initiator side” on page 379

► **Capacity planning basics**

We describe how to calculate the System i CPW and memory you need to support your iSCSI environment, and how to determine the number of target HBAs required.

Refer to 9.11, “Capacity planning for iSCSI” on page 387.

► **iSCSI IP addressing structure**

We explain how IP addressing works in a System i iSCSI network and present a sample addressing schema that you can use.

Refer to 9.12, “IP addressing in a System i iSCSI network” on page 401.

► **Creating an iSCSI connections map**

We show how you can use the QVNIMAP command to retrieve information about the iSCSI network configuration.

Refer to 9.13, “Creating an iSCSI connections map” on page 405.

► **Scaling tasks**

We guide you through the steps necessary to configure the scaled up xSeries and Blade server scenarios described in 9.5, “Scenarios for scaling a hosted xSeries server connection” on page 343 and 9.6, “Scenarios for scaling a hosted Blade server connection” on page 352.

Refer to 9.14, “Scaling tasks” on page 417.

► **Path deployment tasks**

We guide you through the steps necessary to manually deploy storage space links and Virtual Ethernet LAN connections on both the System i side and the hosted server side described in 9.9, “Manual path deployment - target side” on page 377 and 9.10, “Manual path deployment - initiator side” on page 379.

Refer to 9.15, “Path deployment tasks” on page 436.

9.1 Overview

In the overview we introduce the terminology used in the implementation of iSCSI on the System i, and describe the concepts of scaling as they apply to a System i iSCSI network.

We cover the following topics in this section:

- **Terminology**

Here we explain the terms used in this chapter.

Refer to 9.1.1, “Terminology” on page 327.

- **Introduction to scaling the iSCSI Network**

Here we provide an overview of how we go about scaling a System i iSCSI network.

Refer to 9.1.2, “Introduction to scaling the iSCSI network” on page 330.

9.1.1 Terminology

In this chapter, we use diagrams to present the various configurations that can be used to scale the iSCSI network. The diagrams are composed of labelled boxes. Table 9-1 lists each type of label and its corresponding description. If you are not already familiar with the objects and devices whose labels are described in Table 9-1 and how they fit into the iSCSI implementation on System i, refer to the following Web site:

http://www.ibm.com/systems/i/bladecenter/about_integration.html

Table 9-1 Object labels

Label	Full name and description
HBA I HBA	Host bus adapter - In the context of this chapter, the term HBA specifically refers to an iSCSI adapter. The iSCSI HBA in the System i is called the target HBA, and the HBA in the xSeries or Blade server is called the initiator HBA. I HBA is used as an abbreviation for initiator HBA.
NWSD	Network server description object - The i5/OS NWSD object describes the i5/OS environment for an instance of Windows. Among other functions, the NWSD associates an instance of Windows with an xSeries or Blade server and controls the startup and shutdown of the server hardware. There is one NWSD for each instance of Windows that has been installed.
NWSH	Network server host adapter object - The i5/OS NWSH object describes a target HBA. You link a storage space to an NWSH (and thereby to a target HBA) using a storage path. (The term storage path is described in Table 9-2.) You only need to configure one NWSH for each target HBA. Although you can have multiple NWSH objects representing the one physical target HBA there is no point in doing this; it would simply be confusing.
RMTSYS	Remote system object - The i5/OS RMTSYS object describes a hosted xSeries or Blade server. There is always a one to one relationship between a RMTSYS object, which describes the server hardware (xSeries or Blade), and an NWSD, which represents an instance of Windows running on the server.
P	Initiator HBA port - There is one iSCSI port per xSeries initiator HBA, and two ports per Blade initiator HBA.
SRVPRC	Service Processor object - The i5/OS SRVPRC object represents the Service Processor in an xSeries server, or the Management Module in a BladeCenter.

Label	Full name and description
CNNSEC	Communications security object - The i5/OS CNNSEC object is not used in the initial implementation of iSCSI on System i, however it must be configured. In this chapter, the CNNSEC object is usually drawn using dotted lines because it is not directly relevant to the examples provided. It is only shown for completeness. Multiple hosted servers can share the same CNNSEC object.
SP	xSeries Service Processor - This is either a Baseboard Management Controller (BMC) or Remote Support Adapter II (RSA II) in an xSeries server. It enables an xSeries server to be remotely controlled by IBM Director running in the hosting i5/OS partition via a LAN connection.
MM	BladeCenter Management Module - It enables a BladeCenter and its Blade servers to be remotely controlled by IBM Director running in the hosting i5/OS partition via a LAN connection.
LAN	LAN - This represents a System i Ethernet adapter. In the context of this chapter, the Ethernet LAN adapter in the System i is used by IBM Director, running in an i5/OS partition, to communicate with a Service Processor or Management Module to control the operation of a hosted server. Note that this LAN adapter can also be used for normal System i LAN communications.
B	Boot drives - These are the hosted Windows server's C: and D: drive storage spaces. They must both be linked to the Windows server's NWSD using the same storage path.
NB	Non-boot drive - This is an additional storage space or spaces that you can create and link to a Windows server's NWSD after you have performed the initial installation of the hosted server. The initial installation creates the boot drives only.

Throughout this chapter, we use terms that can mean different things depending on the context. In Table 9-2, we define the meaning of some of the common terms used in this chapter.

Table 9-2 Common terms

Term	Description
SCSI	<p>Small computer systems interface - This is a protocol optimized for accessing storage devices, namely disk, tape and optical drives. A computer's storage management subsystem issues SCSI commands to a storage device, which returns the requested data.</p> <p>In the context of this chapter, we often use the term SCSI to refer to disk I/O traffic that is traversing the iSCSI network, as distinct from Virtual Ethernet LAN I/O traffic.</p>
iSCSI	<p>Internet SCSI - As the name implies, Internet SCSI enables SCSI commands and data to be transmitted across the internet. To be more specific, iSCSI is a protocol, which encapsulates SCSI commands and data in TCP/IP packets for transmission across a TCP/IP network, which could be the Internet or any other TCP/IP-capable medium such as an Ethernet LAN. The iSCSI HBAs, which perform the encapsulation, are basically 1 Gb Ethernet adapters, which encapsulate SCSI commands and data in TCP/IP packets. System i supports a hardware only implementation of iSCSI. Other platforms can use a software implementation where the encapsulation is performed in software, and normal Ethernet LAN adapters are used to transmit the data across the TCP/IP network. The hardware implementation of iSCSI is usually preferred for performance reasons.</p>

Term	Description
iSCSI network	An iSCSI network, as implemented on System i, is a switched 1 Gb Ethernet network that connects the iSCSI initiator HBAs installed in the xSeries or Blade servers to the iSCSI target HBAs installed in the System i. We recommend that the iSCSI network is a private Ethernet network (only connects iSCSI HBAs), which is physically secured, but it does not have to be either private or physically secure.
Intranet	In the context of this chapter, an intranet is meant to represent a site LAN and is usually shown separately from the iSCSI network. However, they could be the same network as long as both are running at 1 Gb. We recommend that the intranet and iSCSI networks are kept separate for performance and security reasons. In the System i iSCSI implementation, the intranet is used by IBM Director, running in the hosting i5/OS partition, to communicate with a hosted xSeries server or BladeCenter via its Service Processor or Management Module for management purposes.
Storage space	A storage space is a chunk of i5/OS single level disk storage that Windows sees as a physical disk drive. Each Windows boot or non-boot drive is a storage space in the hosting i5/OS partition. We use the term storage space when we are talking about disk storage from an i5/OS perspective.
Virtual disks	A virtual disk is another name for a storage space. We use the term virtual disk when we are talking about disk storage from a Windows perspective. iSeries Navigator also uses the term virtual disk rather than storage space.
Hosting partition	A hosting partition is an i5/OS partition that is providing the hardware and software resources to support and manage one or more hosted xSeries or Blade servers connected to the iSCSI network.
Hosted server	A hosted server is an xSeries or Blade that is connected to an i5/OS hosting partition via an iSCSI network. The hosted server uses hardware and software resources out of the i5/OS hosting partition.
Integrated server	An integrated server is an IXS, or IXA-connected xSeries server. We use the term integrated server to differentiate IXS/IXA servers from iSCSI-connected servers.
Storage path	A storage path is an access point for a storage device to an NWSH. In other words, a storage path provides a connector for a storage device to link to an NWSH. A storage device usually means a storage space, but it can also refer to removable media (optical or tape). Once a storage device is connected to an NWSH via a storage path, data can be transmitted from the storage device, through the target HBA described by the NWSH, and across the iSCSI network to the hosted server. For a more detailed definition of a storage path, refer to 9.7.2, "Storage paths" on page 364.
IQN	iSCSI qualified name - In the System i iSCSI context this is a unique name that represents a storage path. Initiator IQNs can either be user-defined, or automatically generated by i5/OS. However, unless you have a specific reason, we recommend that you let them be automatically generated. Target IQNs are always automatically generated by i5/OS. In the wider storage context, an IQN is an industry standard term that uniquely represents a storage area network (SAN) interface. Note that IQNs do not apply to Virtual Ethernet LANs. For more information about IQNs, refer to the following Web site: http://www.ietf.org/rfc/rfc3720.txt

Term	Description
Data path	A data path is different from a storage path. A data path is the name given to the connection between a target HBA and an initiator HBA port. Note that, in the initial implementation of iSCSI on System i, disk I/Os for a specific storage space or I/Os for a specific Virtual Ethernet LAN flow over one and only one data path. There is a planned enhancement to the iSCSI implementation on System i to enable multiple data paths to service a single storage space. This capability will be a function of Multipath I/O (MPIO). The process of assigning a storage space (virtual disk) or a Virtual Ethernet LAN to a data path is called path deployment.
Initiator HBA port	An initiator HBA port is a physical iSCSI interface on an initiator HBA in an xSeries or Blade server. We talk about initiator HBA ports rather than initiator HBAs, because you can have one or two ports on one physical initiator HBA adapter in a Blade server. Therefore, we need to distinguish between an initiator HBA and an initiator HBA port. Note that a target HBA (or an initiator HBA for an xSeries server) only ever has one port; therefore, we talk about a target HBA, not a target HBA port.

9.1.2 Introduction to scaling the iSCSI network

Unlike the IXS/IXA architectural model where there is essentially only one object (the network server description or NWSD) that describes the relationship between the integrated Windows server and its hosting partition, the System i iSCSI architecture is not so simple.

A single hosted Windows server connected to an i5/OS partition using iSCSI can contain multiple initiator HBA ports communicating with multiple target HBA ports in the hosting partition. In addition, a single target HBA can support multiple xSeries or Blade servers.

Because the iSCSI implementation on System i is more flexible and more scalable than IXS/IXA, there are additional objects that need to be created to describe the environment. These additional objects include the NWSH, RMTSYS, SRVPRC, and CNNSEC objects that are described in brief in Table 9-1 on page 327.

In this section, we take you through the different ways you can scale the iSCSI network, and describe how to set up the various scaling scenarios.

When we talk about “scaling”, we are essentially talking about two distinct concepts:

- **Increasing the bandwidth of a hosted server connection**

We can increase the bandwidth of a particular hosted server connection by increasing the number of target-initiator HBA data paths between the server and its hosting partition. This enables more SCSI and Virtual Ethernet LAN traffic to flow between the hosted server and its hosting partition. We describe this in 9.2, “Increasing the bandwidth of a hosted server connection” on page 331.

- **Increasing the number of hosted servers**

We can also scale the System i iSCSI network by increasing the number of servers that are hosted by the i5/OS partition. However, we must be careful to maintain adequate bandwidth by installing sufficient target HBAs in the hosting partition to support the I/O requirements of the additional hosted servers. This is described in 9.3, “Increasing the number of hosted servers” on page 340.

9.2 Increasing the bandwidth of a hosted server connection

We cover the following topics in this section:

- **Increasing the bandwidth of a hosted xSeries server connection**

Here we discuss how you can scale up the I/O bandwidth of a hosted xSeries server by adding additional target and initiator HBA ports.

Refer to 9.2.1, “Increasing the bandwidth of a hosted xSeries server connection” on page 331.

- **Increasing the bandwidth of a hosted Blade server connection**

Here we discuss how you can scale up the I/O bandwidth of a hosted Blade server by adding additional target and initiator HBA ports.

Refer to 9.2.2, “Increasing the bandwidth of a hosted Blade server connection” on page 335.

The System i currently supports iSCSI networks running at a speed of 1 Gigabit (Gb). This means that the target and initiator HBAs in a System i iSCSI network have a nominal throughput of 1 Gigabit per second (Gbps).

A dedicated 1 Gb connection will provide adequate bandwidth for a hosted Windows server in most instances. However, for high end Windows servers that are running disk I/O intensive applications such as Microsoft SQL Server, or applications that heavily use the Virtual Ethernet LAN, a 1 Gb connection might not be enough. In these cases, you can increase the bandwidth of the connection between the hosting i5/OS partition and the hosted server by adding additional target HBAs in the hosting partition, and additional initiator HBA ports in the xSeries or Blade server. This enables more I/Os to flow between the hosting i5/OS partition and the hosted Windows server.

You can increase the number of initiator HBA ports in an xSeries server to increase the bandwidth of the iSCSI connection between the hosting i5/OS partition and hosted server by simply adding more iSCSI HBAs (up to four). In the case of a Blade server, you can only install a single initiator HBA adapter, but it can support two iSCSI ports. Note that each of these two ports must be connected to a different Ethernet switch module in the BladeCenter. Ethernet switch modules, which expose the Blade initiator HBA ports to the iSCSI network, are located in bays three and four of the BladeCenter.

Important: The number of target HBAs that a hosted server is communicating with must always be equal to or greater than the number of initiator HBA ports in the hosted server.

Just because you increase the number of initiator HBA ports in a hosted server, it does not mean that the iSCSI network automatically makes use of them. Each storage space and each Virtual Ethernet LAN communicate over a target-initiator HBA port pair. We define a data path as the connection between a target and an initiator HBA port pair over which disk I/Os and/or Virtual Ethernet I/Os flow. The process of allocating a data path for a storage space or Virtual Ethernet LAN to use is called *path deployment*. Data paths can be deployed automatically or manually. We describe the deployment of data paths in 9.7, “Introduction to path deployment” on page 363.

9.2.1 Increasing the bandwidth of a hosted xSeries server connection

You can increase the bandwidth of the connection between a hosted xSeries server and its hosting i5/OS partition by adding target and initiator HBAs as shown in Figure 9-1 on page 332.

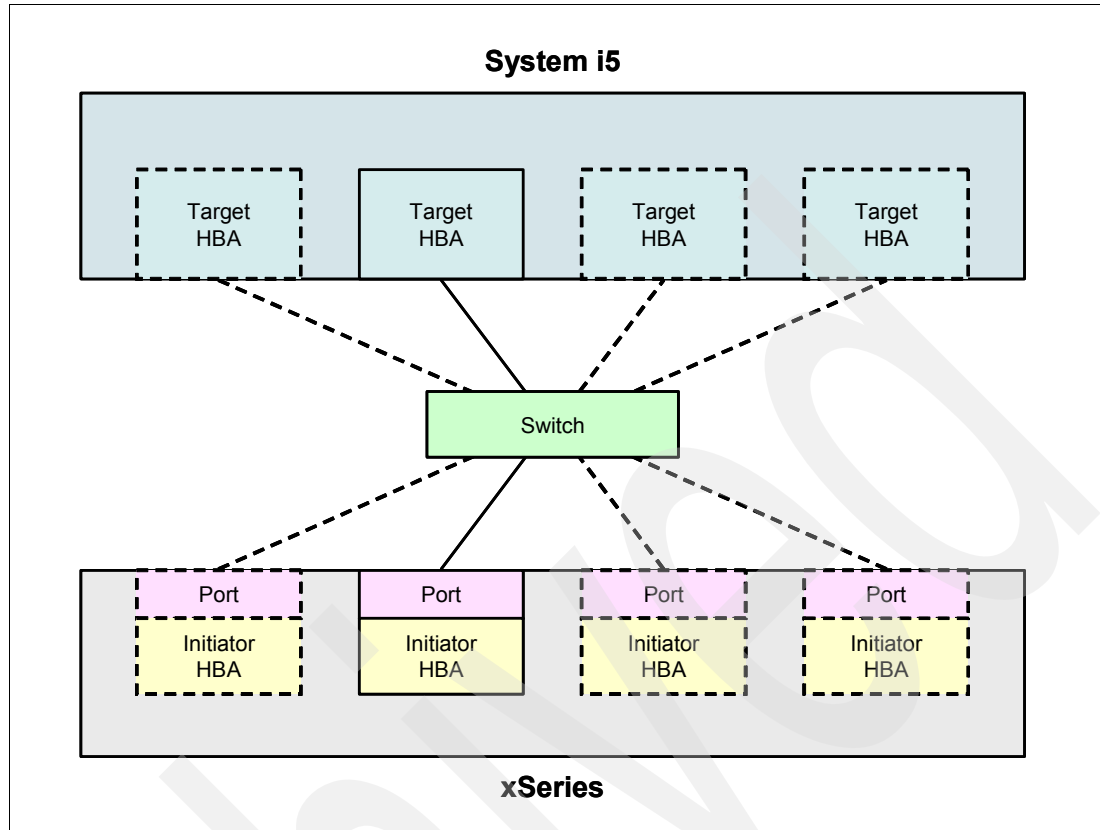


Figure 9-1 Increasing the bandwidth of a hosted xSeries server connection - physical view

In Figure 9-1, note the following points:

- ▶ The minimum number of target and initiator HBA pairs you need to define a data path between an xSeries server and its hosting i5/OS partition is one (shown as solid lines). You can add target and initiator HBAs (shown as dotted lines) to increase the bandwidth of the connection, bearing in mind that the number of target HBAs communicating with the hosted server must be equal to or greater than the number of initiator HBA ports installed in the xSeries server.
- ▶ Each target HBA can establish only one data path to one initiator HBA port, although each initiator HBA port can establish data paths to up to four target HBAs. Therefore, in Figure 9-1, you cannot have more than one initiator HBA port communicating with the same target HBA. This would be meshed data flow, which is not supported. Refer to 9.7.3, “Path deployment considerations” on page 367 for a more detailed discussion of meshed data flow.

We can also depict the target-initiator HBA relationship shown in Figure 9-1 using the i5/OS objects that represent the different components of the iSCSI network. Figure 9-2 on page 333 shows the equivalent object-based view to Figure 9-1.

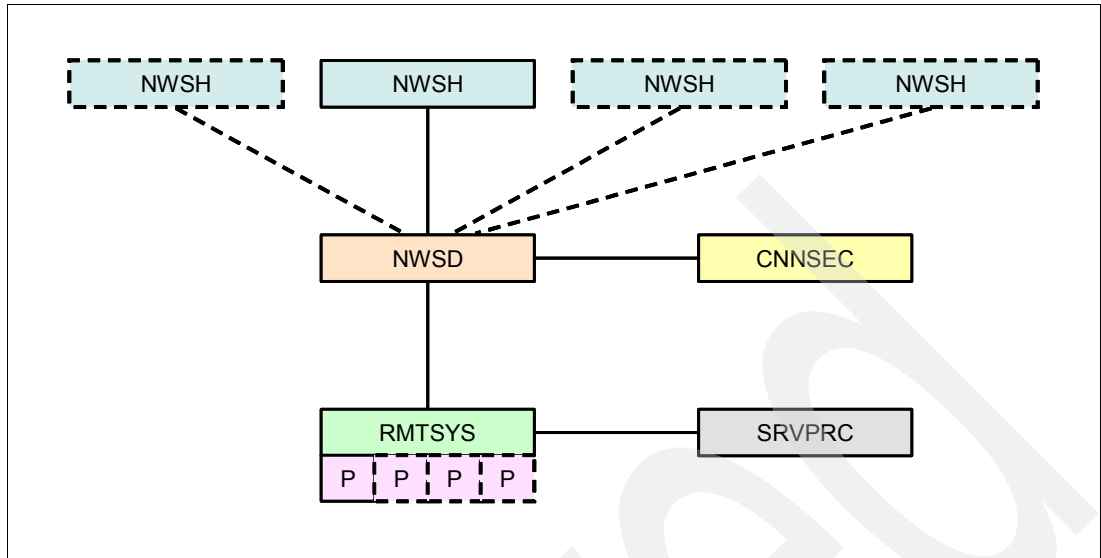


Figure 9-2 Increasing the bandwidth of a hosted xSeries server connection - object-based view

In Figure 9-2, note the following points:

- ▶ Each target HBA is represented by an NWSH object. The NWSH objects are specified in the NWSD.
- ▶ The hosted xSeries server is represented by the NWSD and RMTSYS object pair. There is always a 1 : 1 relationship between the NWSD and RMTSYS objects, that is, an NWSD can point to one and only one RMTSYS.
- ▶ Each P represents an initiator HBA port on the xSeries server. Ports are defined in the RMTSYS object and are, therefore, not shown as separate objects. However, ports are important to the discussion of scalability and path deployment, and we, therefore, include them in our object-based views.

Note: There is one and only one port on each initiator HBA in an xSeries server. Therefore, in the case of an initiator HBA for an xSeries server, we can use the terms initiator HBA and port interchangeably. It is important to understand this because the situation is different for an initiator HBA in a Blade server as discussed in 9.2.2, “Increasing the bandwidth of a hosted Blade server connection” on page 335.

- ▶ The SRVPRC and CNNSEC objects are not relevant to this discussion but are included for completeness. The NWSD points to the CNNSEC object, and the RMTSYS points to the SRVPRC object.

As shown in Figure 9-1 on page 332 and Figure 9-2 on page 333, the maximum number of physical initiator HBAs (ports) that you can install in an xSeries server is four, and the maximum number of target HBAs you can configure to communicate with a hosted xSeries server is also four. In terms of the ratio of target HBAs communicating with initiator HBA ports in an xSeries server, you can have any of the following combinations:

- ▶ Target HBAs : initiator HBAs (ports)
 - 1 : 1
 - 2 : 1
 - 3 : 1
 - 4 : 1
 - 2 : 2
 - 3 : 2
 - 4 : 2
 - 3 : 3
 - 4 : 3
 - 4 : 4

However, to make most efficient use of the target and initiator HBAs' bandwidth, you should keep the number of targets equal to the number of initiator ports for a particular hosted server. There is little point in dedicating four target HBAs to communicate with one initiator HBA port in an xSeries server, because the bandwidth of the single initiator HBA port limits the total throughput between the hosted xSeries server and its hosting partition to 1 Gb.

Things become even more complicated if you are sharing each target HBA among multiple hosted servers. For example, you have four target HBAs and four hosted servers, each with a single initiator HBA port. You define four data paths for each initiator HBA port, one to each of the four target HBAs. While this can be done, you need to be careful because, at the time of writing, there is no redundancy or automatic failover capability on the target side. Therefore, if a target HBA fails, you lose all the data paths that are using that target HBA. In our example, if a target HBA failed, you would lose a data path to all four hosted servers. If the failed data path provided a connection to the boot drives in the hosting partition for any of the hosted servers, those hosted servers will fail. Although the possibility of an HBA failure is very small, it is best to avoid unnecessary data path sharing across target HBAs.

We provide sample configurations of how you can increase the bandwidth of a hosted xSeries server in 9.5, "Scenarios for scaling a hosted xSeries server connection" on page 343.

Once you have added additional target and initiator HBAs to a basic xSeries iSCSI configuration, you might need to redeploy your storage spaces and Virtual Ethernet LANs to make use of the additional bandwidth. You can make changes to the deployment of your storage spaces and Virtual Ethernet LANs on both the hosting partition and hosted server sides as follows:

▶ **Hosting partition side**

On the i5/OS partition side, you must manually deploy storage spaces and Virtual Ethernet LANs to use a specific storage path. In the case of storage spaces, you need to unlink the storage space from one storage path and relink it to another storage path. For Virtual Ethernet LANs, in the NWSD you must specify the NWSH for the Virtual Ethernet LAN to use.

▶ **Hosted server side**

On the xSeries server side, virtual disks and Virtual Ethernet LANs are automatically deployed to initiator HBA ports by default according to a set of IBM proprietary algorithms. However, you can manually deploy virtual disks and Virtual Ethernet LANs to initiator HBA ports if required.

Manual deployment techniques are described in more detail in 9.7, “Introduction to path deployment” on page 363.

9.2.2 Increasing the bandwidth of a hosted Blade server connection

You can increase the bandwidth of the connection between a hosted Blade server and its hosting i5/OS partition by enabling the second port on the initiator HBA.

It is important to understand the difference in port layout between an initiator HBA adapter in an xSeries server and an initiator HBA adapter in a Blade server. The initiator HBA adapter in an xSeries server only ever has one port, but you can install up to four HBA adapters in an xSeries server to give you a total of four ports. In a Blade server, you can only install one initiator HBA adapter. However, the initiator HBA adapter that you install in a Blade server has two ports.

Important: It is important to understand that each of the two ports on an initiator HBA adapter in a Blade server acts as a separate initiator HBA, although there is only one physical HBA adapter in the server. However, to use both ports on an initiator HBA adapter in a Blade server, you need two Ethernet switch or passthru modules in the BladeCenter (one in bay three and one in bay four) because each of the two ports on the iSCSI HBA adapter must be connected to a different switch or passthru module.

Blade initiator HBA ports can only connect to BladeCenter switch (or passthru) modules, and as we have already noted, you need a module in bay three and four of the BladeCenter if you want to make use of both ports on the initiator HBA adapter. You can use a BladeCenter switch module (or modules) as the switch for your iSCSI network, or you can use passthru modules in bays three and four to provide connectivity to an additional external switch if you need extra switching capacity.

Figure 9-3 on page 336 shows one of the ports on the Blade initiator HBA adapter connected to a minimum of one, and a maximum of four, target HBAs.

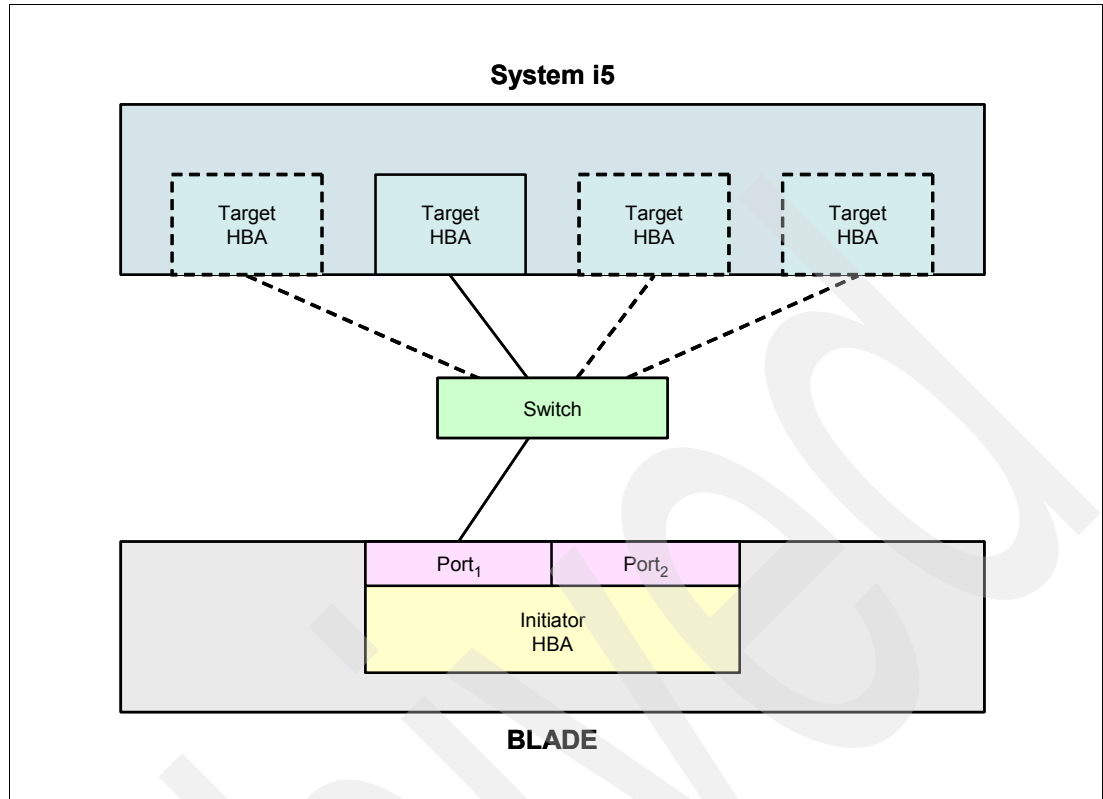


Figure 9-3 Increasing the bandwidth of a hosted Blade connection using one initiator HBA port

In Figure 9-3, note the following points:

- ▶ The minimum number of target and initiator HBA pairs you need to define a data path between a Blade server and its hosting i5/OS partition is one (shown as solid lines). As shown in Figure 9-3 and Figure 9-4 on page 337 you can add target HBAs, and activate the second initiator HBA port, to increase the bandwidth of the connection, bearing in mind that the number of target HBAs communicating with the hosted server must be equal to, or greater than, the number of initiator HBA ports activated in the Blade server.
- ▶ The switch is really a switch module in the BladeCenter, although it could also be a passthru module connected to an external switch. Because there is only one switch module in the BladeCenter, you can only use one of the two ports on the initiator HBA in the Blade server.

Figure 9-4 on page 337 shows both ports on the Blade initiator HBA adapter connected to a minimum of two, and a maximum of four, target HBAs.

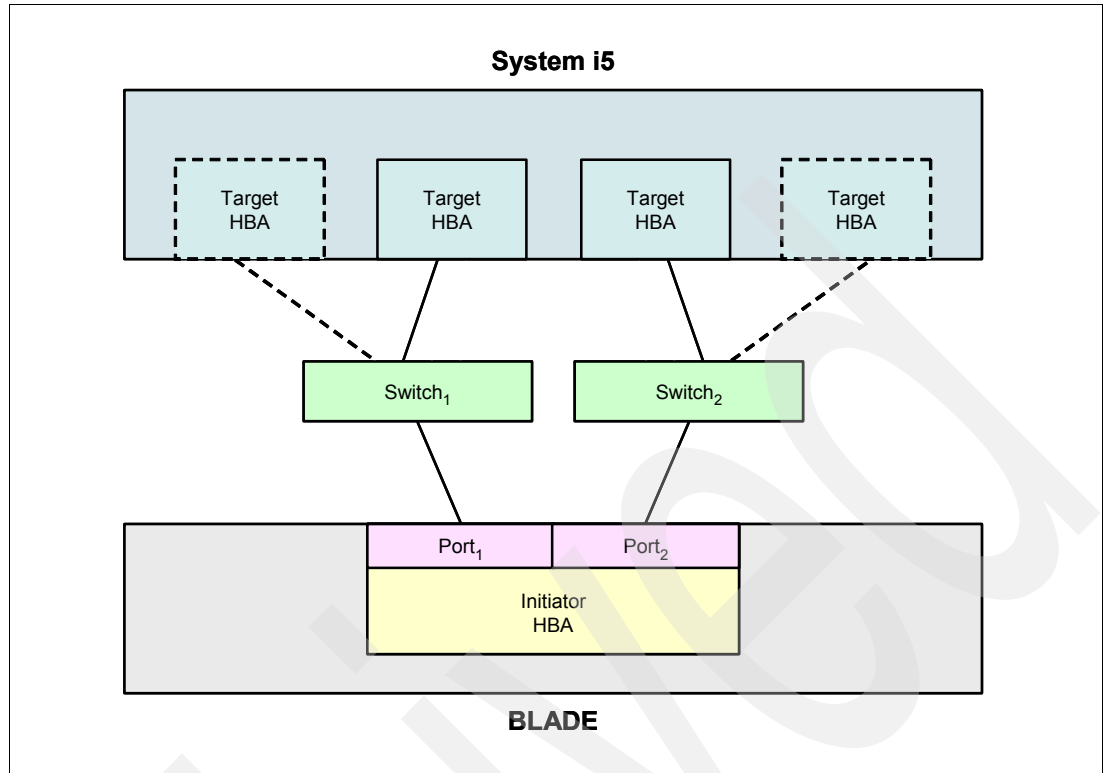


Figure 9-4 Increasing the bandwidth of a hosted Blade connection using both initiator HBA ports

In Figure 9-4, note the following points:

- ▶ The switches in this example are really switch modules in the BladeCenter, although they could also be passthru modules connected to an external switch or switches. Because there are two switch modules, you can now use both of the ports on the initiator HBA adapter in the Blade server. However, each port must connect to separate target HBAs because each port functions as a separate initiator HBA.
- ▶ Each target HBA can establish only one data path to one port on the initiator HBA adapter, although each initiator HBA port can establish data paths to up to four target HBAs. Therefore in Figure 9-4, you cannot have both initiator HBA ports communicating with the same target HBA. This would be meshed data flow, which is not supported. Refer to 9.7.3, "Path deployment considerations" on page 367 for a more detailed discussion of meshed data flow.
- ▶ The data paths between target HBAs and initiator HBA ports are not necessarily spread evenly between the two switches. The way the data paths are deployed can be controlled manually, but there is also code on the Blade server that automatically deploys the data paths from the initiator HBA ports to the target HBAs.

We can also depict the target-initiator HBA relationship shown in Figure 9-3 on page 336 and Figure 9-4 using the i5/OS objects that represent the different components of the iSCSI network. Figure 9-5 on page 338 shows the equivalent object-based view to Figure 9-3 on page 336 and Figure 9-4.

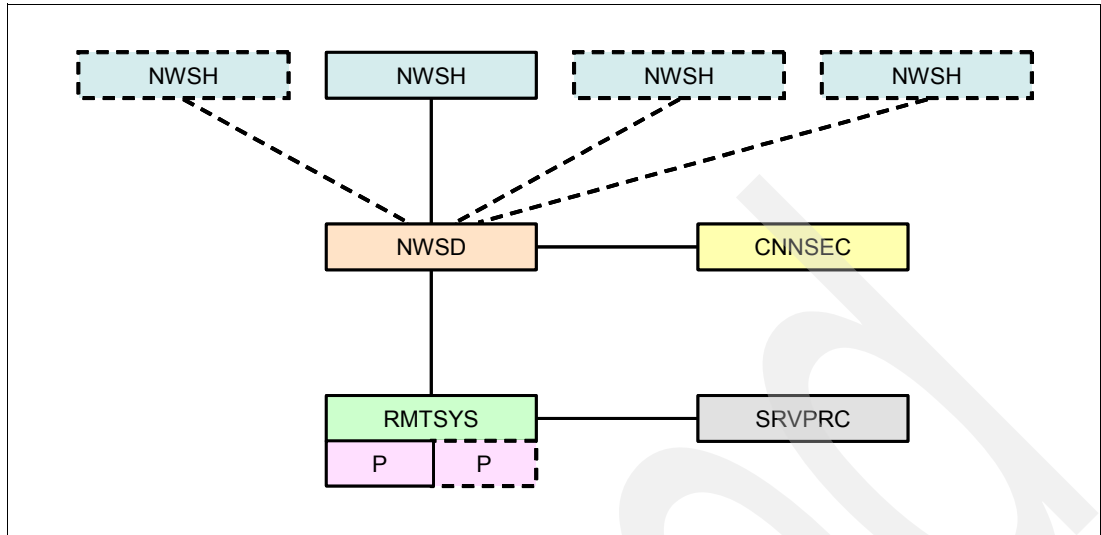


Figure 9-5 Increasing the bandwidth of a hosted Blade connection - object-based view

In Figure 9-5, note the following points:

- ▶ Each target HBA is represented by an NWSH object. The NWSH objects are specified in the NWSD.
- ▶ The hosted Blade server is represented by the NWSD and RMTSYS object pair. There is always a 1 : 1 relationship between the NWSD and RMTSYS objects, that is, an NWSD can point to one and only one RMTSYS.
- ▶ Each P represents an initiator HBA port on the Blade server. Ports are defined in the RMTSYS object and are therefore not shown as separate objects. However, ports are important to the discussion of scalability and path deployment, and we therefore include them in our object-based views.

Note: As noted previously, there are two ports on the initiator HBA adapter in a Blade server. Therefore, in the case of an initiator HBA adapter for a Blade server, we *cannot* use the terms initiator HBA and port interchangeably. Each port on a Blade initiator HBA adapter is internally connected to a different switch (or passthru) module in the BladeCenter.

- ▶ The SRVPRC and CNNSEC objects are not relevant to this discussion but are included for completeness. The NWSD points to the CNNSEC object, and the RMTSYS points to the SRVPRC object.

As shown in Figure 9-3 on page 336, Figure 9-4 on page 337 and Figure 9-5 on page 338, the maximum number of physical initiator HBA adapters you can install in a Blade server is one, but it has two ports, each of which functions as a separate initiator HBA. The maximum number of target HBAs you can configure to communicate with a hosted Blade server is four. In terms of the ratio of target HBAs communicating with initiator HBA ports in a Blade server, you can have any of the following combinations:

- ▶ Target HBAs : initiator HBA ports
 - 1 : 1
 - 2 : 1
 - 3 : 1
 - 4 : 1
 - 2 : 2
 - 3 : 2
 - 4 : 2

However, to make most efficient use of the target and initiator HBAs' bandwidth, you should keep the number of targets equal to the number of initiator ports for a particular hosted server. There is little point in dedicating four target HBAs to communicate with one initiator HBA port in a Blade server because the bandwidth of the single initiator HBA port limits the total throughput between the hosted Blade server and its hosting partition to 1 Gb.

Things become even more complicated if you are sharing each target HBA among multiple hosted servers. For example, you have four target HBAs and four hosted servers, each with a single initiator HBA port. You define four data paths for each initiator HBA port, one to each of the four target HBAs. While this can be done, you need to be careful because, at the time of writing, there is no redundancy or automatic failover capability on the target side. Therefore, if a target HBA fails, you lose all the data paths that are using that target HBA. In our example, if a target HBA failed, you would lose a data path to all four hosted servers. If a failed data path provides a connection to the boot drives in the hosting partition for any of the hosted servers, those hosted servers will fail. Although the possibility of an HBA failure is very small, it is best to avoid unnecessary data path sharing across target HBAs.

We provide sample configurations of how you can increase the bandwidth of a hosted Blade server in 9.6, "Scenarios for scaling a hosted Blade server connection" on page 352.

Once you have added additional target and initiator HBA ports to a basic Blade iSCSI configuration, you might need to redeploy your storage spaces and Virtual Ethernet LANs to make use of the additional bandwidth. You can make changes to the deployment of your storage spaces and Virtual Ethernet LANs on both the hosting partition and hosted server sides as follows:

▶ **Hosting partition side**

On the i5/OS partition side, you must manually deploy storage spaces and Virtual Ethernet LANs to use a specific storage path. In the case of storage spaces, you need to unlink the storage space from one storage path and relink it to another storage path. For Virtual Ethernet LANs, in the NWSD you must specify the NWSH for the Virtual Ethernet LAN to use.

▶ **Hosted server side**

On the Blade server side, virtual disks and Virtual Ethernet LANs are automatically deployed to the initiator HBA ports by default according to a set of IBM proprietary algorithms. However, you can manually deploy virtual disks and Virtual Ethernet LANs to initiator HBA ports if required.

Manual deployment techniques are described in more detail in 9.7, "Introduction to path deployment" on page 363.

9.3 Increasing the number of hosted servers

Here we discuss how you can scale out your iSCSI network by adding additional xSeries and Blade servers.

The maximum number of xSeries and Blade servers in your iSCSI network is no longer constrained by the number of towers you can connect to the System i system unit as with IXA. The limits of an iSCSI network are much less well defined and depend on a number of factors. Although the number of iSCSI HBAs you can install in a particular model of the System i is approximately the same as the combined number of IXSs and IXAs for the same model, there are three factors that make the iSCSI implementation cheaper, more flexible, and more scalable. They are:

- **Server stacking**

The System i currently supports iSCSI networks running at a speed of 1 Gb. This means that a target HBA in a System i iSCSI network has a nominal throughput of 1 Gbps. Depending on the configuration of your hosted servers, and the number of SCSI and Virtual Ethernet LAN I/Os they are generating, a single target HBA might be able to handle the workload of several hosted servers that do not require high bandwidth for their SCSI and Virtual Ethernet LAN traffic. For example, you could share a target HBA among several development and test servers if their workload is light. This is called *server stacking*.

A single target HBA can have concurrent connections to up to eight initiator HBA ports, each in a separate xSeries or Blade server. Note that you need to be very careful when stacking multiple hosted servers on a single target HBA because these servers need to share the 1 Gb bandwidth of the HBA. Carefully assess the bandwidth requirements of the servers you want to attach to ensure that saturation of the target HBA and resulting poor performance do not occur. We provide capacity planning guidelines in 9.11, "Capacity planning for iSCSI" on page 387.

- **Slot density**

The number of slots in the System i system unit and I/O towers that can accommodate an iSCSI HBA is greater than the number of slots in which you can install an IXS for any given model of the System i. Therefore, the cost of System i hardware required to support a hosted server using iSCSI is less on average than that required to support an IXS. Note that the iSCSI HBA that you install in the System i does not require an IOP, that is, it is "IOP-less." This increases the number of slots in an I/O tower in which you can install iSCSI HBAs.

- **More powerful servers**

All Windows servers that you connect to System i using iSCSI are either xSeries or Blades. Either way, they are more powerful than the IXS whose processor GHz is constrained by the power that a PCI slot can supply, and the cooling capability of the I/O tower.

Each model of the System i has a maximum number of iSCSI HBAs that can be accommodated. The limits are documented on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/index.html#support>

Figure 9-6 on page 341 shows the scalability of a single iSCSI target HBA (NWSH) in terms of the maximum number of xSeries and Blade servers that it can communicate with.

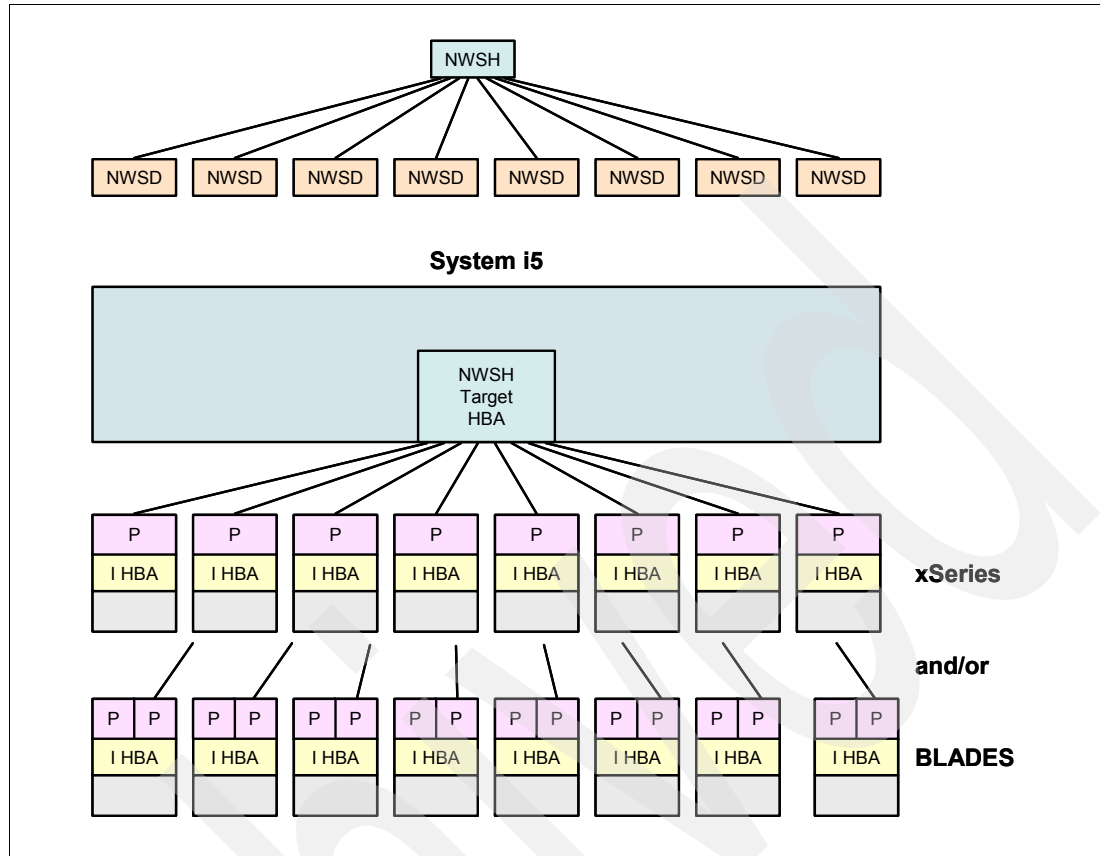


Figure 9-6 Scalability of a single iSCSI target HBA

In Figure 9-6, note the following points:

- ▶ Both the logical (object) relationship and the physical relationship are shown.
- ▶ I HBA stands for initiator HBA, and each I HBA represents a separate xSeries or Blade server. (In this example, we are assuming that only one initiator HBA adapter is installed in each xSeries server.)
- ▶ P refers to a port on the initiator HBA adapter, and to maximize the number of connected Blade servers we only activate one port on each adapter.
- ▶ You can mix and match xSeries and Blade servers on the same target HBA.
- ▶ You can connect up to eight hosted servers to a single target HBA so that they all share the same 1 Gb bandwidth of the iSCSI adapter. However, we do not recommend this because the more hosted servers you have sharing a single target HBA, the more you restrict the bandwidth to any one particular server. This could potentially result in degraded performance due to the bottlenecking of SCSI and/or Virtual Ethernet LAN I/O capacity. For more information about performance, refer to 9.11, “Capacity planning for iSCSI” on page 387.

9.4 iSCSI configuration nomenclature

When discussing iSCSI configurations, it is useful to have a nomenclature that uniquely defines a particular configuration of an xSeries or Blade server connection to the hosting partition.

Knowing this nomenclature is not essential to understanding the configurations described in this chapter; it is simply a useful method of describing a specific xSeries or BladeCenter iSCSI configuration.

One way that an iSCSI configuration can be systematically described is shown in Figure 9-7.

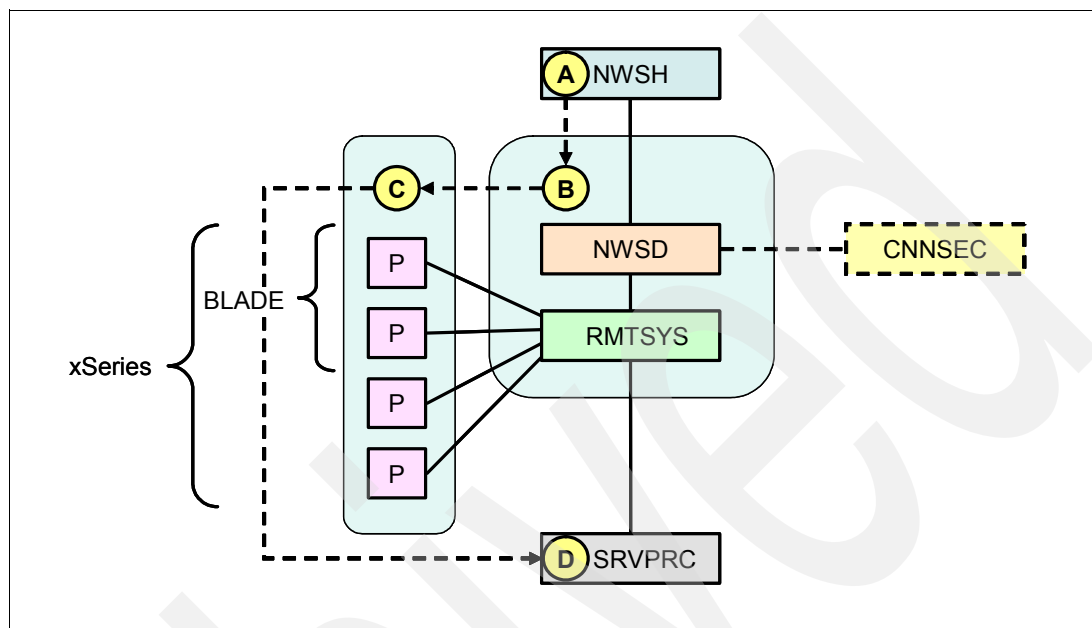


Figure 9-7 Naming an iSCSI configuration

In Figure 9-7, A, B, C, and D represent the numerical relationship between the four significant components of an iSCSI configuration as follows:

- ▶ A represents the number of NWSH objects that an xSeries or Blade server is communicating with.
- ▶ B represents the number of NWSD and RMTSYS objects that is communicating with the NWSH object. The NWSD and RMTSYS objects always exist in pairs, and there is a 1:1 relationship between them. The NWSD describes the i5/OS configuration for a particular instance of Windows, and the RMTSYS object describes the xSeries or Blade server hardware that the instance of Windows is running on. Although they are separate objects, for the purposes of our discussion they form a single unit.
- ▶ C represents the number of initiator HBA ports in the xSeries or Blade server that is being used for communication with the target HBAs in the i5/OS partition.
- ▶ D represents the number of Service Processors (in the case of xSeries) or Management Modules (in the case of BladeCenter).

We write this relationship as A : B : C : D : ? where ? is an X if the hosted server is an xSeries server, or B if the hosted server is a Blade server. We use this nomenclature to describe the configuration examples discussed in this chapter.

Note that you can only configure a single communications security configuration object (CNNSEC) for each NWSD, although the same CNNSEC object can be specified for multiple NWSDs. At the time of writing, the IPsec security protocol is not yet implemented. Therefore, although the CNNSEC object must be created, and specified in the NWSD, it is not used. Thus, it is drawn with a dotted line because it is not relevant to our discussions.

9.5 Scenarios for scaling a hosted xSeries server connection

As described in 9.1.2, “Introduction to scaling the iSCSI network” on page 330, there are essentially two ways that you can scale the xSeries server environment within an iSCSI network:

- ▶ **Increasing the bandwidth of a hosted xSeries server connection**

We can increase the bandwidth for a particular hosted xSeries server by increasing the number of target and initiator HBA connections to that server.

- ▶ **Increasing the number of hosted xSeries servers**

We can increase the number of xSeries servers being hosted by an i5/OS partition, but we must be careful to maintain adequate bandwidth for these servers.

Here we examine the different ways you can scale your xSeries server environment within a System i iSCSI network. Using the nomenclature described in 9.4, “iSCSI configuration nomenclature” on page 341, we cover the following scenarios:

- ▶ 1 : 1 : 1 : 1 : X - The basic iSCSI connection as described on page 343
- ▶ 1 : 2 : 2 : 2 : X - Sharing a target HBA between hosted servers as described on page 345
- ▶ 2 : 1 : 2 : 1 : X - Adding bandwidth to a hosted server as described on page 346
- ▶ 2 : 2 : 2 : 2 : X - Adding another basic iSCSI configuration as described on page 349
- ▶ 2 : 1 : 1 : 1 : X - Splitting the workload between target HBAs as described on page 351

9.5.1 xSeries configuration 1: The basic iSCSI configuration

We identify this configuration as 1 : 1 : 1 : 1 : X.

Description

Figure 9-8 on page 344 shows the simplest iSCSI connection between an xSeries server and its hosting i5/OS partition. We call this the “basic iSCSI configuration” for an xSeries server. It is the basic configuration that we build on to create more complex iSCSI networks. This is what you should end up with after working through the installation procedure documented on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>

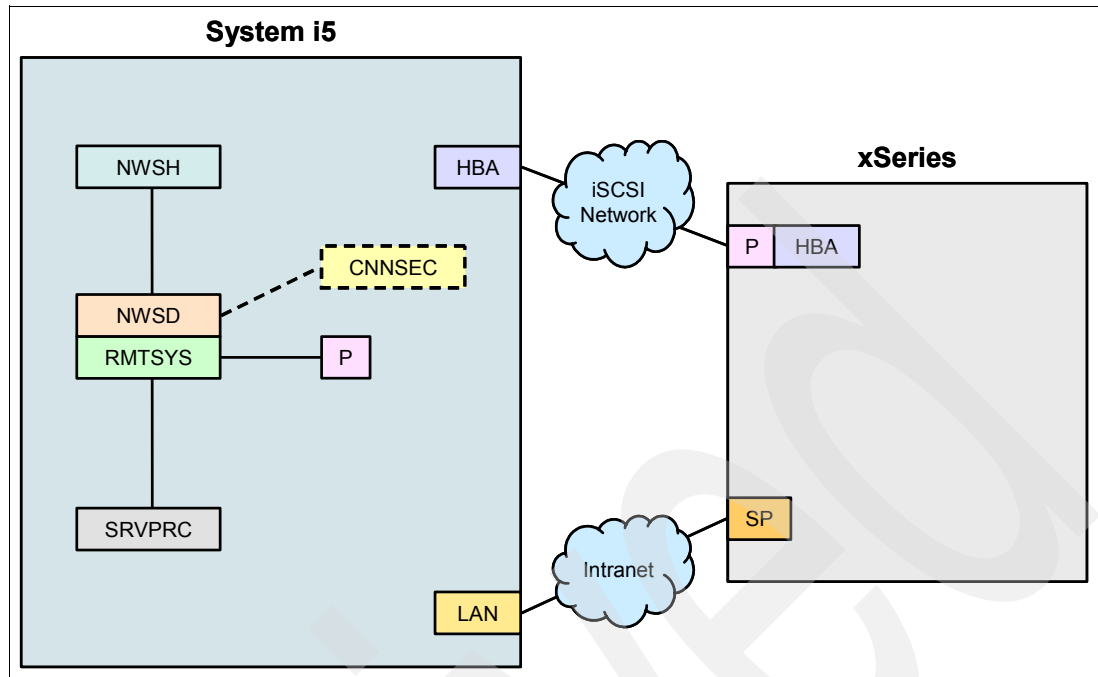


Figure 9-8 Configuration 1 : 1 : 1 : 1 : X

Benefits

This configuration has the following benefits:

- ▶ It is the simplest to configure.
- ▶ It provides good performance in most instances.

Considerations

This configuration has the following considerations:

- ▶ Depending on your application, it might provide too little bandwidth between the xSeries server and its hosting i5/OS partition. For example, if you are running a very disk intensive Windows application such as Microsoft SQL Server, a single iSCSI connection might not provide enough bandwidth to give optimal I/O performance. In this case, you might need to add additional target and initiator HBA ports to prevent the iSCSI network from becoming a bottleneck.
- ▶ Depending on your application, it can provide too much bandwidth between the xSeries server and its hosting partition. If you are running a Windows application that does very little disk access, then you might have excess bandwidth on the iSCSI network. In this case, you might be able to share the target HBA with additional xSeries servers. However, be sure that the capacity of the target HBA exceeds the combined bandwidth requirements of the xSeries servers you are communicating with because poor performance can result if you exceed the capacity of the target HBA. Virtual Ethernet LAN activity also uses bandwidth on the iSCSI network. You need to take this into consideration when deciding on the ratio of target HBAs to initiator HBA ports. For information about capacity planning, refer to 9.11, "Capacity planning for iSCSI" on page 387.

Setup

This is the basic iSCSI configuration which you need to set up before you can configure any of the other xSeries configurations described in this section. To set up this configuration, you need to work through the installation procedure documented on the following Web site:

9.5.2 xSeries configuration 2: Sharing a target HBA between hosted servers

We identify this configuration as 1 : 2 : 2 : 2 : X.

Description

Figure 9-9 shows a configuration where there are two xSeries servers, each with a single initiator HBA port connected to the iSCSI network, communicating with a single target HBA. This configuration is an example of how you can connect multiple xSeries (or Blade) servers (up to eight) to a single target HBA to share its bandwidth.

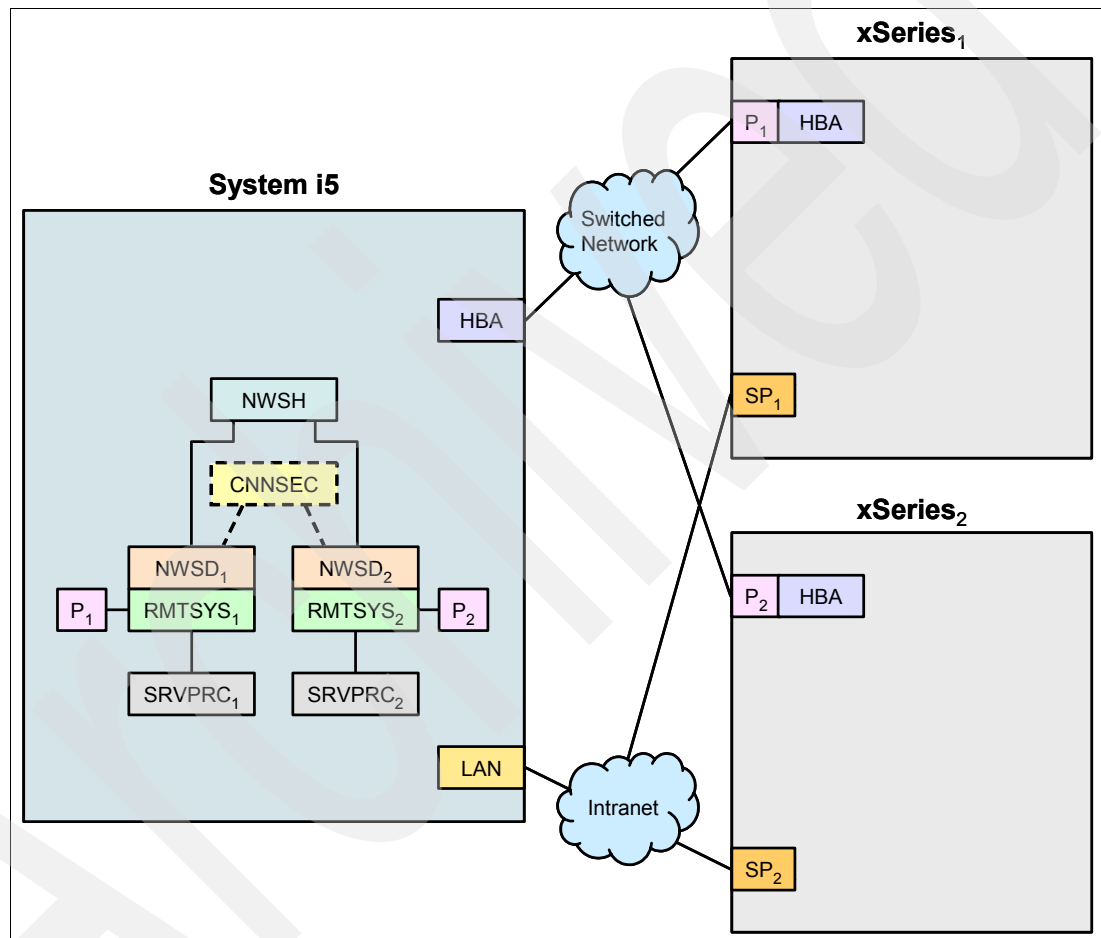


Figure 9-9 Configuration 1 : 2 : 2 : 2 : X

In Figure 9-9, each hosted server has its own set of configuration objects except for the NWSH and CNNSEC objects, which are shared.

Benefits

A single target HBA might be able to handle the workload of several hosted servers that do not require high bandwidth for their SCSI and Virtual Ethernet LAN traffic. For example, you could share a target HBA among several development and test servers if their workload is light.

Considerations

This configuration has the following considerations:

- ▶ If you are considering setting up this configuration, you need to be sure that you have enough bandwidth to support the SCSI and Virtual Ethernet LAN traffic through a single target HBA. Although you can have up to eight xSeries (or Blade) servers connected to a single target HBA, we do not recommend it for performance reasons. The practical limit is determined by the available target HBA bandwidth and the number of I/Os being generated by the hosted servers.
- ▶ There are limits on the number of active SCSI and Virtual Ethernet LAN connections that a single target HBA can support. There are eight file server and eight Virtual Ethernet LAN resource “slots” supported by one NWSH (target HBA). Each file server slot can service all the storage spaces that are linked to the NWSH via the storage path for a particular active server. Each Virtual Ethernet slot can service all the Virtual Ethernet LAN connections that are assigned to the NWSH for a particular active server. The number of available file server and Virtual Ethernet slots limits the number of active hosted servers that can use an NWSH. Note that an active server can use file server and Virtual Ethernet LAN slots on multiple NWSHs.

Setup

To set up this configuration, you need to perform the following tasks:

- ▶ Install the basic xSeries iSCSI configuration as shown in 9.5.1, “xSeries configuration 1: The basic iSCSI configuration” on page 343 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
- ▶ Install a second basic xSeries iSCSI configuration for the second xSeries server but use the same NWSH object that you used to set up the first basic configuration.

9.5.3 xSeries configuration 3: Adding bandwidth to a hosted server

We identify this configuration as 2 : 1 : 2 : 1 : X.

Description

Figure 9-10 on page 347 shows a configuration where you are adding bandwidth to an xSeries server by adding an additional target-initiator HBA pair. This effectively doubles the bandwidth between the xSeries server and i5/OS partition. You might want to add bandwidth if the xSeries server is running an I/O intensive application such as Microsoft SQL server, or if you are heavily using a Virtual Ethernet LAN connection on the xSeries server.

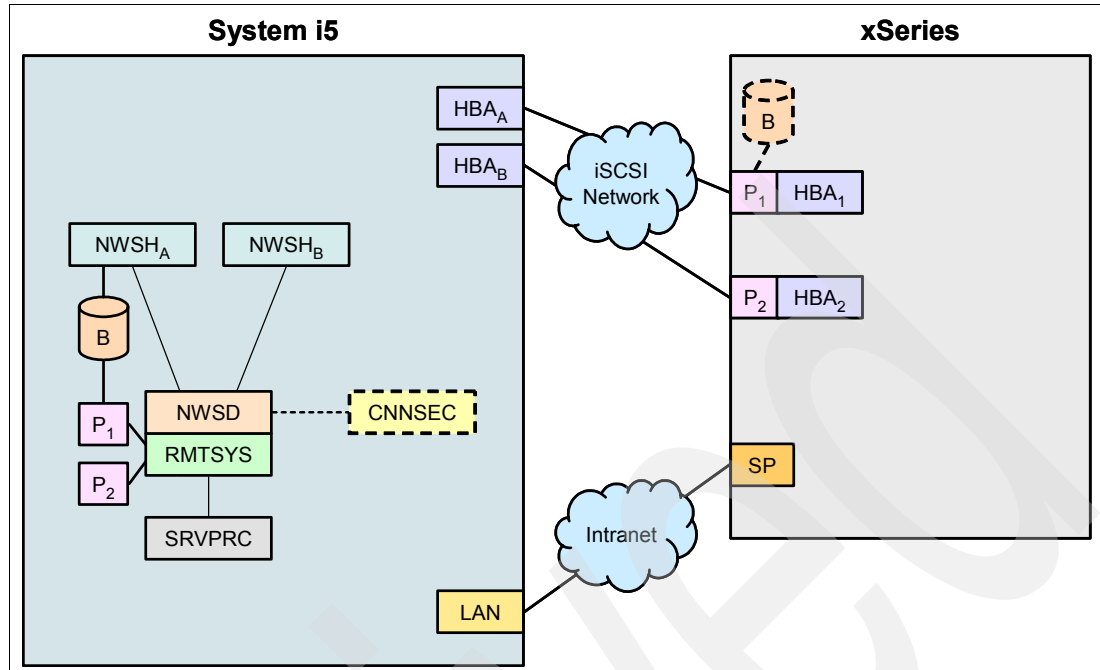


Figure 9-10 Configuration 2 : 1 : 2 : 1 : X

In Figure 9-10, note the following points:

- ▶ To connect the xSeries server to two target HBAs, you specify two storage paths in the NWSD. Each storage path corresponds to a separate target HBA (NWSH). The number of target HBAs a hosted server is communicating with must always be equal to or greater than the number of initiator HBA ports in the hosted server.
- ▶ You must manually assign storage space links and Virtual Ethernet LAN connections to one target HBA or the other. There is no autodeployment on the target side.
- ▶ You can specify those initiator HBA ports that you do *not* want virtual disks to be automatically deployed to.
- ▶ You can manually assign Virtual Ethernet LAN lines to one initiator HBA port or the other, or let them deploy automatically.
- ▶ One of the initiator HBAs' ports on the xSeries server must be identified as the boot port. The *boot port* is the initiator HBA port where the hosted server looks for a bootable disk drive, or in this case, a virtual disk containing the Windows operating system. For example, in Figure 9-10, P₁ has been specified as the boot port in the RMTSYS configuration object. If there were only one initiator HBA port in the hosted server, this port would automatically be deployed as the boot port.

Benefits

This configuration has the following benefits:

- ▶ It enables you to increase the effective bandwidth for disk accesses and Virtual Ethernet LAN communications between the hosted server and its hosting partition. The bandwidth of a single target-initiator port pair might not be enough to support the SCSI and/or Virtual Ethernet LAN traffic being generated by the hosted server. You can install additional target-initiator port pairs (up to a total of four) to provide even greater bandwidth.
- ▶ It enables you to split the SCSI and Virtual Ethernet traffic from each initiator HBA port across two target HBAs by dividing up the storage space links and Virtual Ethernet LAN

connections for the xSeries server between the two NWSHs. This enables you to balance the workload from the two initiator ports across the two target HBAs for best performance.

Considerations

This configuration has the following considerations:

- ▶ In order to balance the workload across the two initiator HBA ports in the xSeries server, you need to manually assign the virtual disks and Virtual Ethernet LAN connections to one port or the other. However, if you do nothing, autodeployment automatically splits the virtual disk links and Virtual Ethernet LAN connections between the initiator HBA ports. Autodeployment does not attempt to do any workload balancing.
- ▶ You also need to decide how you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload. Note that storage spaces and Virtual Ethernet LANs must always be manually deployed on the target side if you have two or more target HBAs. There is no autodeployment.
- ▶ The number of slots in the System i in which you can install target HBAs is limited. By dedicating multiple target HBA slots to a single hosted server, you might be limiting the number of servers you can host out of the i5/OS partition.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic xSeries iSCSI configuration as shown in 9.5.1, “xSeries configuration 1: The basic iSCSI configuration” on page 343 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.
4. Create a storage path for the hosted server using the second target HBA as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
5. Install a second initiator HBA in the xSeries server as described in 9.14.4, “Installing an additional initiator HBA in an xSeries server” on page 425.
6. Configure the second initiator HBA in the xSeries server as described in 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426.

This task is composed of the following subtasks:

- a. First, you must configure the additional initiator HBA port as a non-boot port *or* a boot port:
 - To configure the additional *initiator HBA port* as a non-*boot port*, refer to 9.14.6, “Configuring the additional initiator HBA port as a non-boot port” on page 426.
 - To configure the additional *initiator HBA port* as the boot port, refer to 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429.
 - b. Secondly, you must update the RMTSYS configuration object with the boot port and iSCSI IP addressing information:
 - To update the RMTSYS configuration object, refer to 9.14.8, “Updating the RMTSYS configuration object” on page 433.
7. Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.

8. Redeploy Virtual Ethernet LANs between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.
9. (Optional) Redeploy virtual disks between initiator HBA ports as described in 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
10. (Optional) Redeploy Virtual Ethernet LANs between initiator HBA ports as described in 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

9.5.4 xSeries configuration 4: Adding another basic iSCSI configuration

We identify this configuration as 2 : 2 : 2 : 2 : X.

Description

Figure 9-11 shows a configuration where you are scaling out by adding an additional xSeries server and matching target HBA.

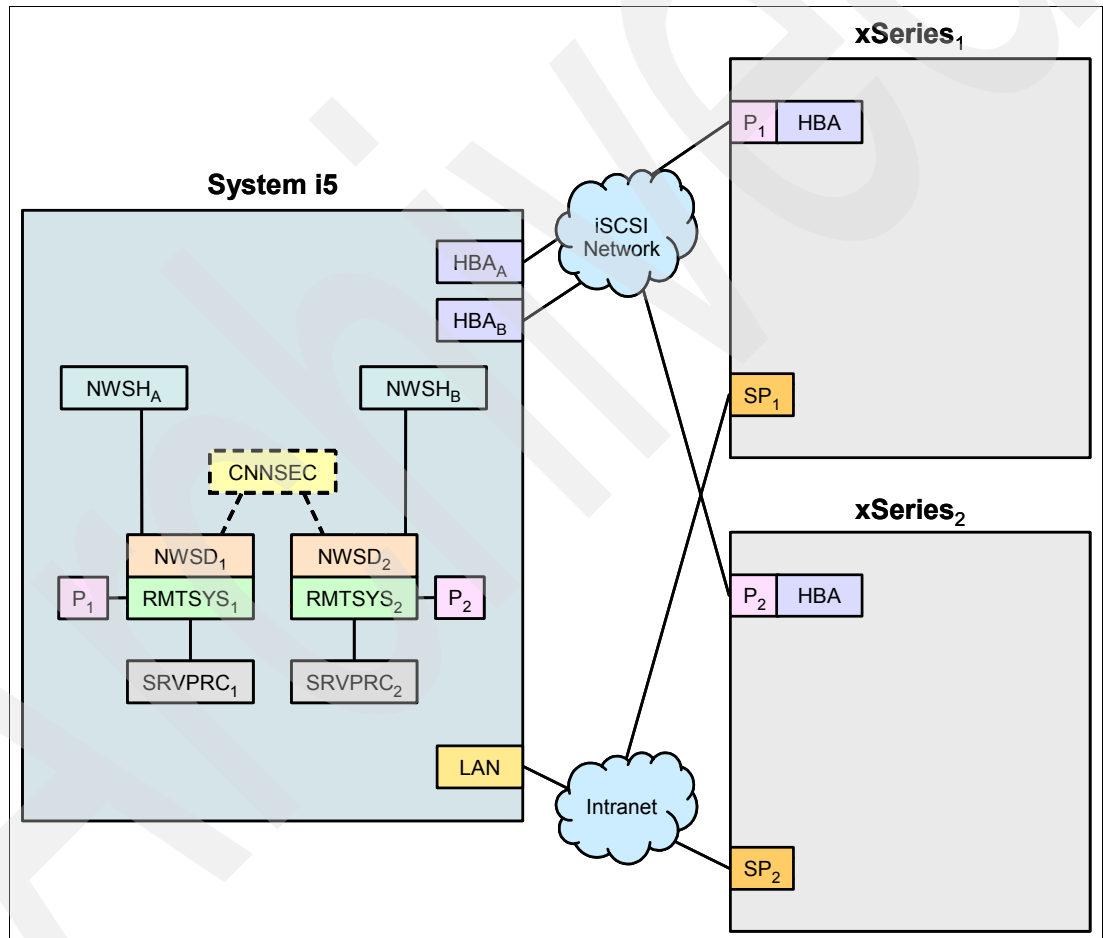


Figure 9-11 Configuration 2 : 2 : 2 : 2 : X

In Figure 9-11, note the following points:

- ▶ You are simply adding another basic iSCSI configuration as described in 9.5.1, “xSeries configuration 1: The basic iSCSI configuration” on page 343.
- ▶ Although not specifically covered in this configuration, you could connect each xSeries server to both target HBAs by specifying a storage path for each NWSH in both NWSDs. In this case, you must manually assign storage space links and Virtual Ethernet LAN

connections to one target HBA or the other. There is no autodeployment on the target side.

Benefits

This configuration has the following benefits:

- ▶ It provides an increase in bandwidth in the hosting partition to offset the connection of an additional hosted server.
- ▶ It enables you to split the SCSI and Virtual Ethernet LAN traffic from each xSeries server across two target HBAs by spreading the storage space links and Virtual Ethernet LAN connections for each hosted server across the two NWSHs. This enables you to balance the workload from the two xSeries servers between the two target HBAs for best performance.

Considerations

This configuration has the following considerations:

- ▶ The number of slots in the System i in which you can install target HBAs is limited. If your hosted servers are only performing very light workloads, you might not need to install an additional target HBA to provide adequate bandwidth to both hosted servers.
- ▶ In order to use the two target HBAs effectively, you need to decide if you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload.
- ▶ While probably not a concern for two target HBAs, the more you split the storage space links and Virtual Ethernet LAN connections between target HBAs the greater the risk of losing multiple hosted servers if a target HBA fails. The reason for this is that, at the time of writing, there is no storage path redundancy or automatic failover capability available across target HBAs. If a target HBA fails, you lose all the storage paths and Virtual Ethernet LAN connections that were using that target HBA. Although the possibility of an HBA failure is very small, it is best to minimize target HBA sharing between hosted servers.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic iSCSI configuration as shown in 9.5.1, “xSeries configuration 1: The basic iSCSI configuration” on page 343 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.
4. Install a second basic xSeries iSCSI configuration for the second hosted server using the second target HBA.
5. (Optional) Create storage paths for both hosted servers using both target HBAs as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
6. (Optional) Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
7. (Optional) Redeploy Virtual Ethernet LANs between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.

9.5.5 xSeries configuration 5: Splitting the workload between target HBAs

We identify this configuration as 2 : 1 : 1 : 1 : X.

Description

Figure 9-12 shows a configuration where you are splitting storage spaces and/or Virtual Ethernet LANs between two target HBAs.

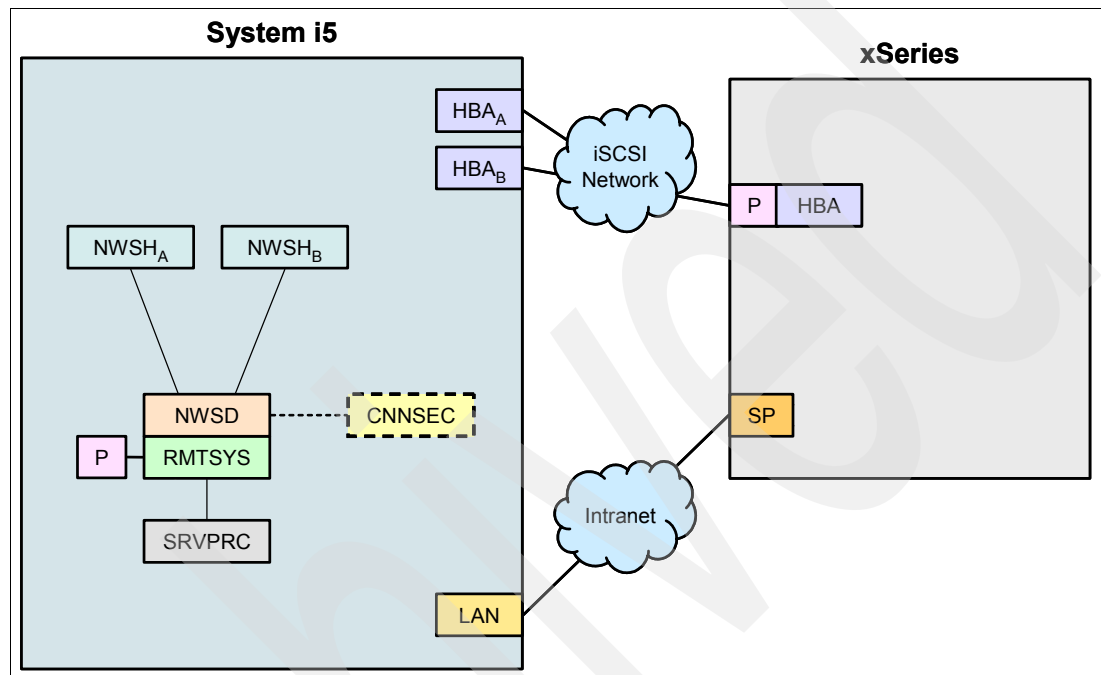


Figure 9-12 Configuration 2 : 1 : 1 : 1 : X

In Figure 9-12, this configuration is only of benefit if you split storage space links and/or Virtual Ethernet LANs between the two target HBAs. In this case, you must manually assign storage space links and Virtual Ethernet LAN connections to one target HBA or the other. There is no autodeployment on the target side.

Benefits

This configuration has the benefit that, although there is only one initiator HBA port in the hosted server, you can increase the throughput of the connection between the xSeries server and i5/OS partition by splitting storage space links and Virtual Ethernet LAN connections between the two target HBAs. This configuration can provide improved performance compared with a single target HBA.

Considerations

This configuration has the following considerations:

- ▶ Although the throughput between the hosted server and hosting partition is greater than when using a single target HBA, the performance is still constrained by the single initiator HBA port in the xSeries server. Therefore, the benefit is not as great as having two initiator HBA ports installed in the xSeries server.
- ▶ The number of slots in the System i in which you can install target HBAs is limited. If your hosted server is only performing very light workloads, you might not need to install an additional target HBA to provide adequate bandwidth to the hosted server.

- You need to decide how you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload. Note that storage spaces and Virtual Ethernet LANs must always be manually deployed on the target side if you have two or more target HBAs connected to a single hosted server. There is no autodeployment.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic xSeries iSCSI configuration as shown in 9.5.1, “xSeries configuration 1: The basic iSCSI configuration” on page 343 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.
4. Create a storage path for the hosted server using the second target HBA as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
5. Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
6. Redeploy Virtual Ethernet LAN connections between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.

9.6 Scenarios for scaling a hosted Blade server connection

As described in 9.1.2, “Introduction to scaling the iSCSI network” on page 330, there are essentially two ways that you can scale the Blade server environment within an iSCSI network:

► Increasing the bandwidth of a hosted Blade server connection

We can increase the bandwidth for a particular hosted Blade server by increasing the number of target and initiator HBA connections to that server.

► Increasing the number of hosted Blade servers

We can increase the number of Blade servers being hosted by an i5/OS partition, but we must be careful to maintain adequate bandwidth for these servers.

Here we examine the different ways that you can scale your Blade server environment within a System i iSCSI network. Using the nomenclature described in 9.4, “iSCSI configuration nomenclature” on page 341, we cover the following scenarios:

- 1 : 1 : 1 : 1 : B - The basic iSCSI connection as described on page 353
- 1 : 2 : 2 : 1 : B - Sharing a target HBA between hosted servers as described on page 354
- 2 : 1 : 2 : 1 : B - Adding bandwidth to a hosted server as described on page 355
- 2 : 2 : 2 : 1 : B - Adding another basic iSCSI configuration as described on page 358
- 2 : 2 : 4 : 1 : B - Splitting the workload across multiple data paths as described on page 360

9.6.1 Blade configuration 1: The basic iSCSI configuration

We identify this configuration as 1 : 1 : 1 : 1 : B.

Description

Figure 9-13 shows the simplest iSCSI connection between a Blade server and its hosting i5/OS partition. We call this the “basic iSCSI configuration” for a Blade server. It is the basic configuration that we build on to create more complex iSCSI networks. This is what you should end up with after working through the installation procedure documented on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>

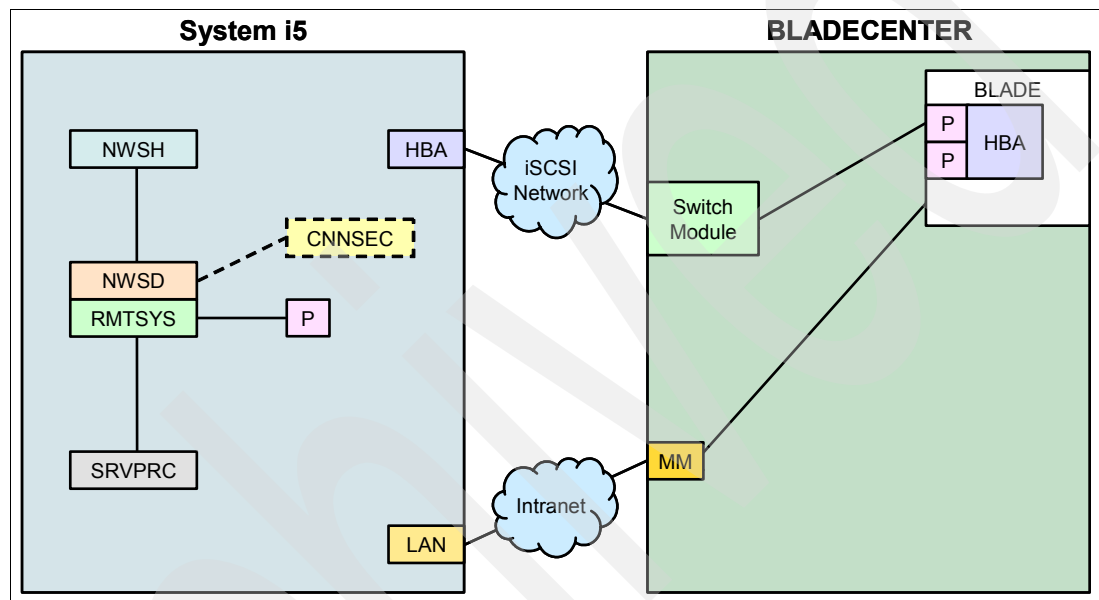


Figure 9-13 Configuration 1 : 1 : 1 : 1 : B

In Figure 9-13 the second port on the initiator HBA cannot be used in this configuration because it would require a second target HBA to be installed in the i5/OS partition. Also, the second initiator HBA port requires the installation of a second switch module in bay four of the BladeCenter.

Benefits

This configuration has the following benefits:

- ▶ It is the simplest to configure.
- ▶ It provides good performance in most instances.

Considerations

This configuration has the following considerations:

- ▶ It might provide too little bandwidth between the Blade server and its hosting i5/OS partition, depending on your application. For example, if you are running a very disk intensive Windows application such as Microsoft SQL Server, a single iSCSI connection might not provide enough bandwidth to give optimal I/O performance. In this case, you might need to add additional target and initiator HBA ports to prevent the iSCSI network from becoming a bottleneck.
- ▶ It might provide too much bandwidth between the Blade server and its hosting partition, depending on your application. If you are running a Windows application that does very

little disk access, then you can have excess bandwidth on the iSCSI network. In this case, you might be able to share the target HBA with additional Blade servers. However, be sure that the capacity of the target HBA exceeds the combined bandwidth requirements of the Blade servers you are communicating with, because very poor performance can result if you exceed the capacity of the target HBA. Virtual Ethernet LAN activity also uses bandwidth on the iSCSI network. You need to take this into consideration when deciding on the ratio of target HBAs to initiator HBA ports. For information about capacity planning, refer to 9.11, “Capacity planning for iSCSI” on page 387.

Setup

This is the basic Blade iSCSI connection, which you need to set up before you can configure any of the other Blade server configurations described in this chapter. To set up this configuration, you need to work through the installation procedure documented on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>

9.6.2 Blade configuration 2: Sharing a target HBA between hosted servers

We identify this configuration as 1 : 2 : 2 : 1 : B.

Description

Figure 9-14 shows a configuration where there are two Blade servers, each with one port on the initiator HBA connected to the iSCSI network, communicating with a single target HBA. This configuration is an example of how you can connect multiple Blade (or xSeries) servers (up to eight) to a single target HBA to share its bandwidth.

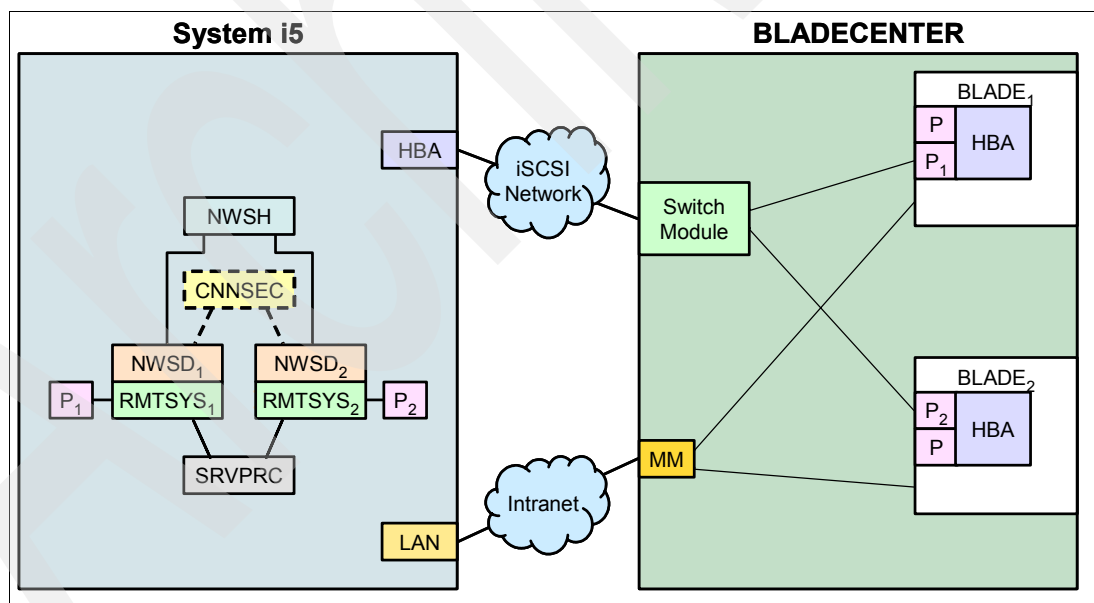


Figure 9-14 Configuration 1 : 2 : 2 : 1 : B

In Figure 9-14, note the following points:

- Each hosted server has its own set of configuration objects except for the NWSH and CNNSEC objects, which are shared.
- The second port on each of the initiator HBAs cannot be used in this configuration because it would require a second target HBA to be installed in the i5/OS partition. Also,

the second port requires the installation of a second switch module in bay four of the BladeCenter.

Benefits

A single target HBA might be able to handle the workload of several hosted servers that do not require high bandwidth for their SCSI and Virtual Ethernet LAN traffic. For example, you could share a target HBA among several development and test servers if their workload is light.

Considerations

This configuration has the following considerations:

- ▶ If you are considering setting up this configuration, you need to be sure that you have enough bandwidth to support the SCSI and Virtual Ethernet LAN traffic through a single target HBA. Although you can have up to eight Blade (or xSeries) servers connected to a single target HBA, we do not recommend it for performance reasons. The practical limit is determined by the available target HBA bandwidth and the number of I/Os that are generated by the hosted servers.
- ▶ There are limits on the number of active virtual disk and Virtual Ethernet LAN connections that a single target HBA can support. There are eight file server and eight Virtual Ethernet LAN resource “slots” supported by one NWSH (target HBA). Each file server slot can service all the storage spaces that are linked to the NWSH via the storage path for a particular active server. Each Virtual Ethernet slot can service all the Virtual Ethernet LAN connections that are assigned to the NWSH for a particular active server. The number of available file server and Virtual Ethernet slots limits the number of active hosted servers that can use an NWSH. Note that an active server can use file server and Virtual Ethernet LAN slots on multiple NWSHs.

Setup

To set up this configuration, you need to perform the following tasks:

- ▶ Install the basic Blade iSCSI configuration as shown in 9.6.1, “Blade configuration 1: The basic iSCSI configuration” on page 353 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
- ▶ Install a second basic Blade iSCSI configuration for the second Blade server but use the same NWSH object that you used to set up the first basic configuration.

9.6.3 Blade configuration 3: Adding bandwidth to a hosted server

We identify this configuration as 2 : 1 : 2 : 1 : B.

Description

Figure 9-15 on page 356 shows a configuration where you are adding bandwidth to a Blade server by adding an additional target-initiator HBA pair. This effectively doubles the bandwidth between the Blade server and i5/OS partition. You might want to add bandwidth if the Blade server is running an I/O intensive application such as Microsoft SQL server, or if you are heavily using a Virtual Ethernet LAN connection on the Blade server.

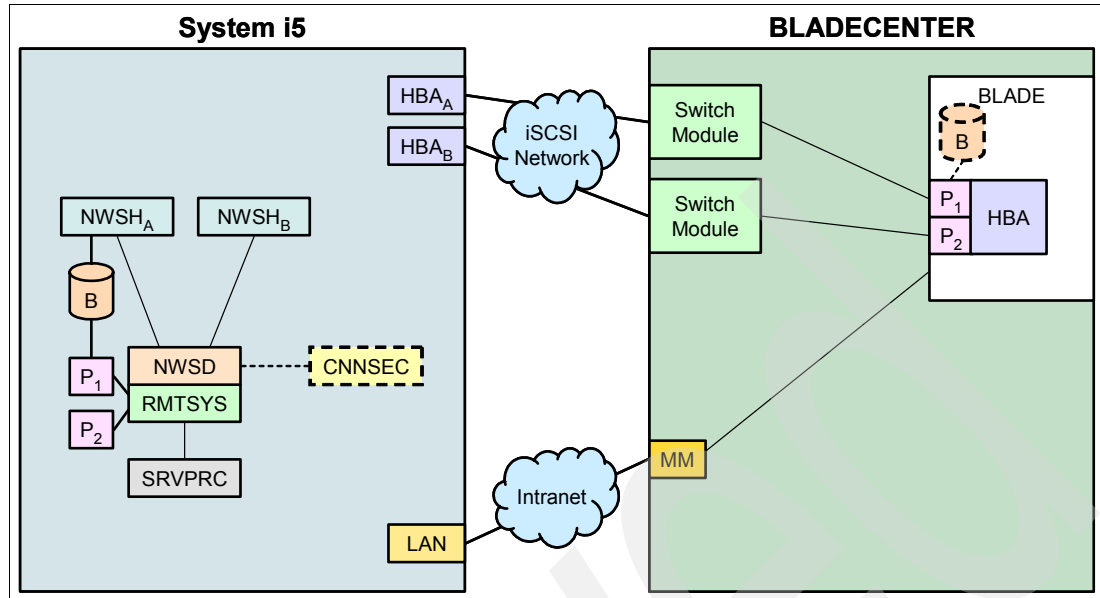


Figure 9-15 Configuration 2 : 1 : 2 : 1 : B

In Figure 9-15, note the following points:

- ▶ To connect the Blade server to two target HBAs, you specify two storage paths in the NWSD. Each storage path corresponds to a separate target HBA (NWSH). The number of target HBAs a hosted server is communicating with must always be equal to or greater than the number of initiator HBA ports in the hosted server.
- ▶ You must manually assign storage space links and Virtual Ethernet LAN connections to one target HBA or the other. There is no autodeployment on the target side.
- ▶ To use both ports on the initiator HBA in the Blade server, you must first install a second switch module in bay four of the BladeCenter.
- ▶ You can specify those initiator HBA ports that you do *not* want virtual disks to be automatically deployed to.
- ▶ You can manually assign Virtual Ethernet LAN lines to one initiator HBA port or the other, or let them deploy automatically.
- ▶ One of the initiator HBA's ports on the Blade server must be identified as the boot port. The boot port is the initiator HBA port where the hosted server looks for a bootable disk drive, or in this case, a virtual disk containing the Windows operating system. For example, in Figure 9-15, P₁ has been specified as the boot port in the RMTSYS configuration object. If there were only one connected initiator HBA port in the hosted server, this port would automatically be deployed as the boot port.

Benefits

This configuration has the following benefits:

- ▶ It enables you to increase the effective bandwidth for disk accesses and Virtual Ethernet LAN communications between the hosted server and its hosting partition. The bandwidth of a single target-initiator port pair might not be enough to support the SCSI and/or Virtual Ethernet LAN traffic that is generated by the hosted server. You can make the second port on the initiator HBA available to the Blade server by installing a second switch module in bay four of the BladeCenter.
- ▶ It enables you to split the SCSI and Virtual Ethernet LAN traffic from each initiator HBA port across two target HBAs by dividing up the storage space links and Virtual Ethernet

LAN connections for the Blade server between the two NWSHs. This enables you to balance the workload from the two initiator ports across the two target HBAs for the best performance.

Considerations

This configuration has the following considerations:

- ▶ On a Blade server, you can only install one initiator HBA, but it provides two ports, each of which functions as a separate initiator HBA. In this case, there is little benefit in installing additional target HBAs to communicate with this Blade server, because the throughput will be limited by the two initiator HBA ports. Contrast this with an xSeries server in which you can install up to four initiator HBAs (one port per HBA).
- ▶ In order to balance the workload across the two initiator HBA ports in the Blade server, you need to manually assign the virtual disks and Virtual Ethernet LAN connections to one port or the other. However, if you do nothing, autodeployment automatically splits the virtual disk links and Virtual Ethernet LAN connections between the initiator HBA ports. Autodeployment does not attempt to do any workload balancing.
- ▶ You need to decide how you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload. Note that storage spaces and Virtual Ethernet LANs must always be manually deployed on the target side if you have two or more target HBAs connected to a single hosted server. There is no autodeployment.
- ▶ The number of slots in the System i in which you can install target HBAs is limited. By dedicating multiple target HBA slots to a single hosted server, you might be limiting the number of servers you can host out of the i5/OS partition.
- ▶ To make use of the second port on the initiator HBA port in the Blade server, you need to install a second switch module in bay four of the BladeCenter. This is the only way to connect to the second port.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic Blade iSCSI configuration as shown in 9.6.1, “Blade configuration 1: The basic iSCSI configuration” on page 353 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.
4. Create a storage path for the hosted server using the second target HBA as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
5. Install a second switch (or passthru) module in bay four of the BladeCenter to support the second port on the initiator HBA.
6. Configure the second initiator HBA port in the Blade server as described in 9.14.9, “Configuring the second initiator HBA port in a Blade server” on page 436. This section describes some minor differences between configuring the second initiator HBA port in a Blade server compared with an xSeries server. Because the differences are minor, you are redirected to 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426.

Configuring the second initiator HBA port is composed of the following subtasks:

- a. First, you must configure the additional initiator HBA port as a non-boot port *or* a boot port:
 - To configure the additional initiator HBA port as a non-boot port, refer to 9.14.6, “Configuring the additional initiator HBA port as a non-boot port” on page 426.
 - To configure the additional initiator HBA port as the boot port, refer to 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429.
 - b. Secondly, you must update the RMTSYS configuration object with the boot port and iSCSI IP addressing information:
 - To update the RMTSYS configuration object, refer to 9.14.8, “Updating the RMTSYS configuration object” on page 433.
7. Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
 8. Redeploy Virtual Ethernet LANs between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.
 9. (Optional) Redeploy virtual disks between initiator HBA ports as described in 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
 10. (Optional) Redeploy Virtual Ethernet LANs between initiator HBA ports as described in 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

9.6.4 Blade configuration 4: Adding another basic iSCSI configuration

We identify this configuration as 2 : 2 : 2 : 1 : B.

Description

Figure 9-16 shows a configuration where you are scaling out by adding an additional Blade server and matching target HBA.

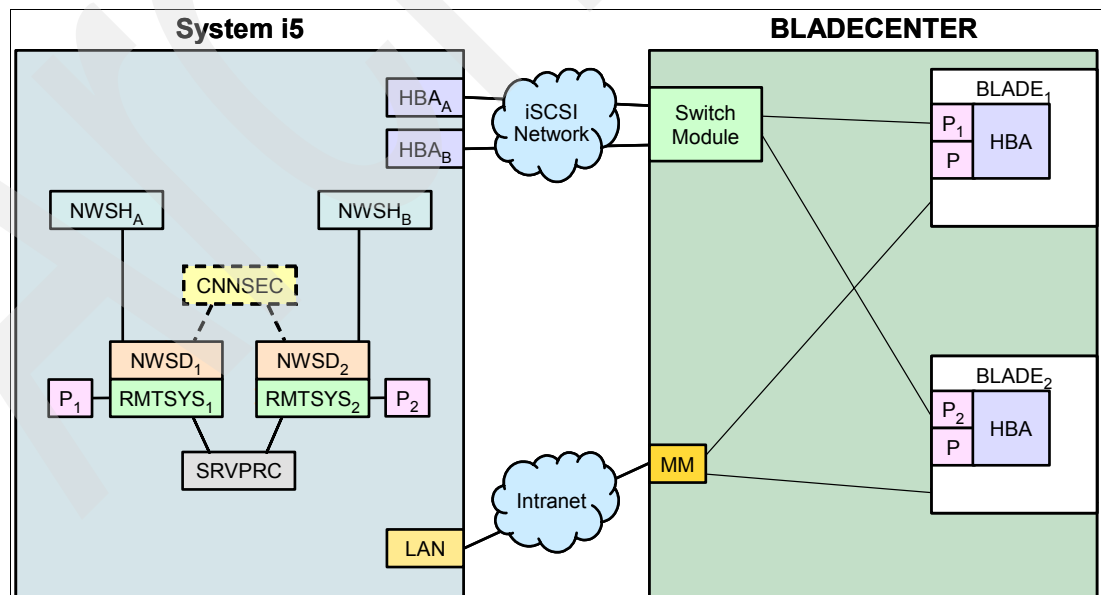


Figure 9-16 Configuration 2 : 2 : 2 : 1 : B

In Figure 9-16 on page 358, note the following points:

- ▶ You are simply adding another basic iSCSI configuration as described in 9.6.1, “Blade configuration 1: The basic iSCSI configuration” on page 353.
- ▶ The second port on each of the initiator HBAs (which must be connected to a second switch module in bay four of the BladeCenter) is not being used.
- ▶ Although not specifically covered in this configuration, you could connect each Blade server to both target HBAs by specifying a storage path for each NWSH in both NWSDs. In this case, you must manually assign storage space links and Virtual Ethernet LAN connections to one target HBA or the other. There is no autodeployment on the target side.

Benefits

This configuration has the following benefits:

- ▶ It provides an increase in bandwidth in the hosting partition to offset the connection of an additional hosted server.
- ▶ It enables you to split the SCSI and Virtual Ethernet LAN traffic from each Blade server across two target HBAs by dividing up the storage space links and Virtual Ethernet LAN connections for each hosted server between the two corresponding NWSHs. This enables you to balance the workload from the two Blade servers across the two target HBAs for best performance.

Considerations

This configuration has the following considerations:

- ▶ The number of slots in the System i in which you can install target HBAs is limited. If your hosted servers are only performing very light workloads, you might not need to install an additional target HBA to provide adequate bandwidth to both hosted servers.
- ▶ In order to use the two target HBAs effectively, you need to decide if you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload.
- ▶ While probably not a concern for two target HBAs, the more you split the storage space links and Virtual Ethernet LAN connections between target HBAs, the greater the risk of losing multiple hosted servers if a target HBA fails. The reason for this is that, at the time of writing, there is no storage path redundancy or automatic failover capability available across target HBAs. If a target HBA fails, you lose all the storage paths and Virtual Ethernet LAN connections that were using that target HBA. Although the possibility of an HBA failure is very small, it is best to minimize target HBA sharing between hosted servers.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic Blade iSCSI configuration as shown in 9.6.1, “Blade configuration 1: The basic iSCSI configuration” on page 353 by working through the installation procedure documented on the following Web site:
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.

4. Install a second basic Blade iSCSI configuration for the second hosted server using the second target HBA.
5. (Optional) Create storage paths for both hosted servers using both target HBAs as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
6. (Optional) Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
7. (Optional) Redeploy Virtual Ethernet LANs between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.

9.6.5 Blade configuration 5: Splitting the workload across multiple data paths

We identify this configuration as 2 : 2 : 4 : 1 : B.

Description

Figure 9-17 shows a configuration where you are not only splitting the workload on each of two Blade servers across two initiator HBA ports (as in 9.6.3, “Blade configuration 3: Adding bandwidth to a hosted server” on page 355), but you are also mixing and matching the storage space links and Virtual Ethernet LAN connections between two target HBAs. This configuration enables you to perform a measure of workload balancing across the target and initiator HBA ports.

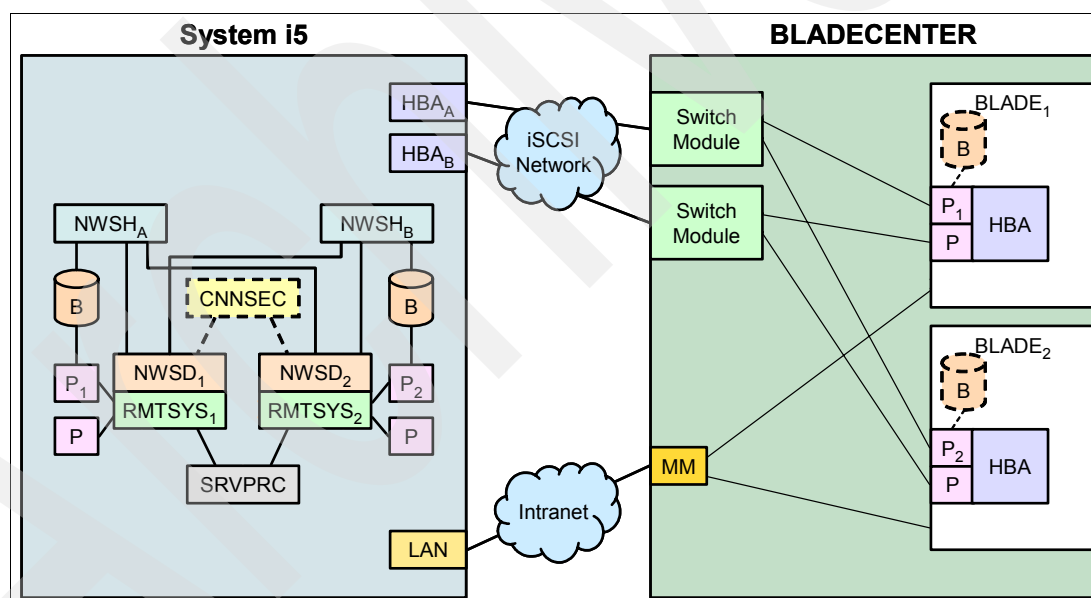


Figure 9-17 Configuration 2 : 2 : 4 : 1 : B

In Figure 9-17, note the following points:

- ▶ To connect a Blade server to two target HBAs, you specify two storage paths in the NWSH. Each storage path corresponds to a separate NWSH (target HBA). The number of target HBAs a hosted server is communicating with must always be equal to or greater than the number of initiator HBA ports in the hosted server.
- ▶ You must manually assign storage space links and Virtual Ethernet LAN connections to one target HBA or the other. There is no autodeployment on the target side.
- ▶ To use both ports on an initiator HBA in a Blade server, you must first install a second switch module in bay four of the BladeCenter.

- ▶ You can specify those initiator HBA ports that you do *not* want virtual disks to be automatically deployed to.
- ▶ You can manually assign Virtual Ethernet LAN lines to one initiator HBA port or the other, or let them deploy automatically.
- ▶ One of the initiator HBA's ports on each Blade server must be identified as the boot port. The boot port is the initiator HBA port where the hosted server looks for a bootable disk drive, or in this case, a virtual disk containing the Windows operating system. For example, in Figure 9-17 on page 360, P₁ and P₂ have been specified as the boot ports in the RMTSYS configuration objects. If there were only one connected initiator HBA port in the hosted server, this port would automatically be deployed as the boot port.

Benefits

This configuration has the following benefits:

- ▶ It provides the most flexibility in managing bandwidth across multiple target HBAs.
- ▶ It enables you to increase the effective bandwidth for SCSI and Virtual Ethernet LAN traffic between the hosted servers and their hosting partition. The bandwidth of a single target-initiator port pair might not be enough to support the SCSI and/or Virtual Ethernet LAN traffic generated by the hosted server. You can make the second port on the initiator HBAs available to the Blade servers by installing a second switch module in bay four of the BladeCenter.
- ▶ It enables you to split the SCSI and Virtual Ethernet traffic from each initiator HBA port across two target HBAs by dividing up the storage space links and Virtual Ethernet LAN connections for each hosted server between the two NWSHs. This enables you to balance the workload from the two Blade servers across the two target HBAs for best performance. You can also split the SCSI and Virtual Ethernet LAN traffic across both initiator HBAs in the Blade servers.

Considerations

This configuration has the following considerations:

- ▶ In order to balance the workload across the two initiator HBA ports in the Blade server, you need to manually assign the virtual disks and Virtual Ethernet LAN connections between them. However, if you do nothing, autodeployment automatically splits the virtual disk links and Virtual Ethernet LAN connections between the initiator HBA ports. Autodeployment does not attempt to do any workload balancing.
- ▶ You need to decide how you want to split the SCSI and Virtual Ethernet LAN traffic between the two target HBAs to balance the workload. Note that storage spaces and Virtual Ethernet LANs must always be manually deployed on the target side if you have two or more target HBAs connected to a single hosted server. There is no autodeployment.
- ▶ You need to purchase a second switch module for bay four of the BladeCenter to support this configuration.

Setup

To set up this configuration, you need to perform the following tasks:

1. Install the basic Blade iSCSI configuration as shown in 9.6.1, “Blade configuration 1: The basic iSCSI configuration” on page 353 by working through the installation procedure documented on the following Web site.
<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>
2. Install a second target HBA in the hosting partition as described in 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.
3. Configure the second target HBA as described in 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.
4. Install a second basic Blade iSCSI configuration for the second hosted server using the second target HBA.
5. Create storage paths for both hosted servers using both target HBAs as described in 9.14.3, “Creating a storage path using the additional target HBA” on page 423.
6. Install a second switch (or passthru) module in bay four of the BladeCenter to support the second port on the initiator HBAs.
7. Configure the second initiator HBA port in the Blade servers as described in 9.14.9, “Configuring the second initiator HBA port in a Blade server” on page 436. This describes some minor differences between configuring the second initiator HBA port in a Blade server compared with an xSeries server. Because the differences are minor, you are redirected to 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426.

Configuring the second initiator HBA port is composed of the following subtasks:

- a. First, you must configure the additional initiator HBA port as a non-boot port *or* a boot port:
 - To configure the additional initiator HBA port as a non-boot port, refer to 9.14.6, “Configuring the additional initiator HBA port as a non-boot port” on page 426.
 - To configure the additional initiator HBA port as the boot port, refer to 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429.
 - b. Secondly, you must update the RMTSYS configuration object with the boot port and iSCSI IP addressing information:
 - To update the RMTSYS configuration object, refer to 9.14.8, “Updating the RMTSYS configuration object” on page 433.
8. (Optional) Redeploy storage spaces between target HBAs as described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
 9. (Optional) Redeploy Virtual Ethernet LANs between target HBAs as described in 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.
 10. (Optional) Redeploy virtual disks between initiator HBA ports as described in 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
 11. (Optional) Redeploy Virtual Ethernet LANs between initiator HBA ports as described in 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

9.7 Introduction to path deployment

We cover the following topics in this section:

- **Overview**

Here we provide an introduction to how data paths between target and initiator HBA ports are established.

Refer to 9.7.1, “Overview” on page 363.

- **Storage paths**

Here we explain the concept of a storage path and describe special types of storage paths.

Refer to 9.7.2, “Storage paths” on page 364.

- **Path deployment considerations**

Here we discuss current limitations of the System i iSCSI architecture that you should be aware of.

Refer to 9.7.3, “Path deployment considerations” on page 367.

9.7.1 Overview

There are two types of I/Os that flow across the SCSI network between target HBAs and initiator HBAs:

- **SCSI I/Os**

SCSI I/Os flow between the storage spaces in the i5/OS hosting partition and hosted servers.

- **Virtual Ethernet LAN I/Os**

Virtual Ethernet LAN I/Os flow between virtual TCP/IP endpoints in the i5/OS hosting partition and hosted servers.

Both types of data flows are connection-oriented, that is, a data path is established when the hosted server starts up and is maintained until the hosted server is shut down.

Important: Path deployment is the process by which a data path is established between a target HBA and an initiator HBA port. Such a data path can be used to transmit SCSI I/Os or Virtual Ethernet LAN I/Os. Path deployment is important because it controls how a hosted server's SCSI and Virtual Ethernet I/Os are routed over the iSCSI network between target and initiator HBA ports.

Servers that require high I/O bandwidth might require more than one target-initiator HBA port pair to handle the workload. You can segment this even further by identifying which virtual disks and Virtual Ethernet LANs require high bandwidth and which ones do not. For example, you can dedicate an iSCSI HBA to a virtual disk that needs high bandwidth, and share another iSCSI HBA among virtual disks or other hosted servers that do not require high bandwidth.

You can really only take practical advantage of path deployment if you have created extra storage spaces and/or Virtual Ethernet LANs in addition to the boot drives and point-to-point Virtual Ethernet LAN that are created during installation of Windows on the hosted server. This is because each storage space (virtual disk) and Virtual Ethernet LAN can only be assigned to one HBA port. Note that the boot drives and point-to-point Virtual Ethernet LAN use the same target HBA and initiator HBA port by default. Therefore, to make practical use

of multiple target and initiator HBA ports, you need to create additional storage spaces and/or Virtual Ethernet LANs to assign to these extra HBAs.

At the time of writing, there is no dynamic data path switching or failover. A future enhancement is planned that will allow you to assign a single storage space to multiple target HBAs for improved performance and dynamic failover capabilities. This new function (called *Multipath I/O*) is referred to in some System i documentation, but it is not available in the initial implementation of iSCSI on System i.

Before we continue the discussion of path deployment, we need to understand how path deployment takes place. Path deployment can occur automatically or manually as follows:

► **Automatic path deployment**

Automatic path deployment (also called *autodeployment*) occurs on the initiator side only, and only when you do not choose to manually select a path. IBM proprietary algorithms, running in Windows, allocate initiator HBA ports to Virtual Ethernet LANs and non-boot virtual disks. Autodeployment occurs independently for SCSI and Virtual Ethernet LAN connections. Note that autodeployment does not attempt to perform any workload balancing, although it does try to balance the number of connections evenly across the initiator HBA ports.

For a discussion and examples of automatic path deployment, refer to 9.8, “Automatic path deployment” on page 369.

► **Manual path deployment**

Manual path deployment can occur on both the target and initiator sides of the iSCSI network. In fact, all storage space links and Virtual Ethernet LAN connections on the target side *must* be deployed manually, except for the C: and D: drives and the point-to-point Virtual Ethernet LAN, which are automatically deployed when the server is created.

Manual path deployment means that you specify which target or initiator HBA port a virtual disk or Virtual Ethernet LAN should use. Therefore, manual deployment is really only meaningful when you have two or more target HBAs configured for connection to a hosted server, *and* two or more initiator HBA ports installed and configured in the hosted server. For a single target-initiator HBA pair (basic iSCSI configuration), there is obviously only one data path that can be used by iSCSI and Virtual Ethernet LAN data flowing between the target HBA and initiator HBA port. Therefore, manual deployment is of no benefit because there is no choice of paths for the data.

You can use manual path deployment to do a measure of workload balancing if you know the number of I/Os being generated by each virtual disk or Virtual Ethernet LAN. Manual deployment enables you to assign virtual disks and Virtual Ethernet LANs to target and initiator HBA ports so that the I/Os are spread evenly over all HBAs.

For a discussion and examples of manual deployment on the target side, refer to 9.9, “Manual path deployment - target side” on page 377.

For a discussion and examples of manual deployment on the initiator side, refer to 9.10, “Manual path deployment - initiator side” on page 379.

9.7.2 Storage paths

We often use the terms storage path and NWSH interchangeably, however, there is an important distinction. A *storage path* is an access point for a storage device to connect to an NWSH. In other words, a storage path provides a connector for a storage device to link to an NWSH. A storage device usually means a storage space, but it can also refer to removable media (optical or tape). Note that storage paths are only applicable to storage devices, not Virtual Ethernet LANs. Once a storage device is connected to an NWSH via a storage path,

data can be transmitted from the storage device, through the target HBA described by the NWSH, and across the iSCSI network to the hosted server.

A storage path is unique to an NWSD in that each NWSH can have only one storage path defined for a particular NWSD. However, an NWSD can contain multiple storage paths (up to four), each representing an access point to a different NWSH.

In summary, a storage “path” can be regarded as an “access point” for a storage device to connect to an NWSH.

We say that a storage space is *linked* to an NWSD via a storage path, whereas a Virtual Ethernet LAN is *connected* to an NWSD via an NWSH. You can link a storage space using one of up to four storage paths, where each storage path points to a specific NWSH. You can connect a Virtual Ethernet LAN directly to one of up to four NWSHs. The NWSH objects that you use to link storage spaces and to connect Virtual Ethernet LANs can be the same set of four NWSHs or a different set of four. In other words, your storage spaces and Virtual Ethernet LANs can use up to eight different NWSHs, where each NWSH describes a different target HBA.

You can link only storage spaces to an NWSH via a storage path, only Virtual Ethernet LANs to an NWSH, or a combination of both storage spaces and Virtual Ethernet LANs to an NWSH. There is a limit of 64 storage spaces that you can link to an NWSD via a single storage path (NWSH), and five Virtual Ethernet LANs that you can connect to an NWSD via a single NWSH.

There are three special types of storage paths that you need to be aware of:

- ▶ The boot path
- ▶ The default (storage) path for disk drives
- ▶ The removable media path

These storage paths are described in this section.

The boot path

In terms of path deployment, the boot path is a special case.

As part of the setup of the hosted server, you use the CTRL-Q utility to select the initiator HBA port used to boot the operating system (Windows in our case). This port is called the boot port. The boot port is connected through the iSCSI network to the Windows server's C: and D: drive storage spaces in the hosting i5/OS partition. When Windows is starting up, it looks for a boot drive (the system drive) on the boot port. After Windows has booted from the boot port, the selected initiator HBA port continues to provide a connection to the system drive storage space in the hosting partition until the server is shut down. The steps you need to perform to change the boot port are described in 9.15.3, “Changing the boot port” on page 443.

The boot path is established between the boot port on the initiator HBA and a target HBA. The boot drives (C: and D:) must both be linked to the same target HBA (NWSH), and you specify the NWSH to which you want the boot drives linked when you run the INSWNTSVR command. This is because Windows needs to know where to find the boot drives. If there were any ambiguity, Windows might not be able to boot up.

At the i5/OS end of the boot path, you can change the target HBA (and therefore the boot path) by linking the boot drives (C: and D:) to a different NWSH than the default NWSH you specified in the INSWNTSVR command. However, you first need to specify the new storage path for the NWSH you are linking to in the NWSD. The NWSD ties the boot path to a hosted

server using the NWSH object that the boot drives are linked to, and the RMTSYS configuration object that describes the hosted server.

Figure 9-18 shows:

- ▶ Two storage paths created for the NWSD Kpax066.
- ▶ The current Default path for disk drives is Nh1 in07 but you could change it to Nh1 in06 (only when the server is shut down).
- ▶ The Default removable media path (for shared System i tape and optical drives) is also Nh1 in07.

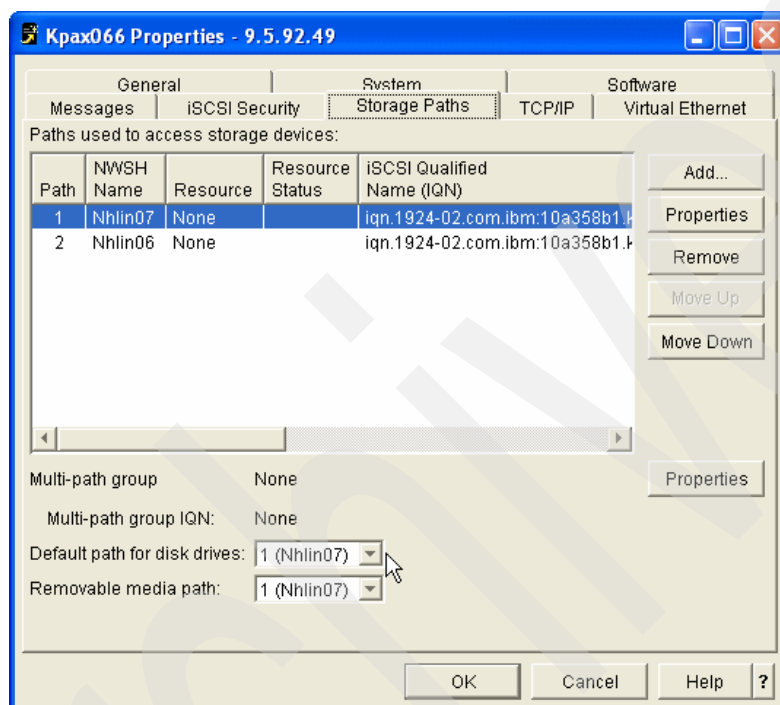


Figure 9-18 Default boot path

The default path for disk drives

The default path for disk drives is the storage path that is used by default to link storage spaces to an NWSD. In other words, when you create a new storage space (virtual disk drive), and you wish to make it available to a Windows server, you first need to link the storage space to the NWSD that controls the Windows server. You also need to tell i5/OS which target HBA will be used to access the storage space. You do this by specifying a storage path when you link the storage space to the NWSD. The default path for disk drives is the storage path used to link storage spaces to the NWSD, unless you specify a different storage path. As shown in Figure 9-18, you can change the default path for disk drives to any storage path that has been assigned to the NWSD.

If you have multiple storage paths specified in the NWSD for a Windows server, it is a good idea to not link all the storage spaces for a Windows server using the same storage path. If you spread the storage spaces across multiple storage paths, you are also spreading the disk I/Os across multiple target HBAs and thereby making better use of the available bandwidth. However, we recommend that you keep the number of storage paths to the minimum necessary to adequately spread the load.

The default path for disk drives defaults to the storage path that you specify in the INSWNTSVR command when you create the Windows server.

The removable media path

One of the many benefits of the System i Windows integration technology is the ability for an integrated Windows server to use native System i hardware devices such as tape and optical drives. This can provide Windows with access to high speed and high capacity storage devices that might otherwise not be available for Windows to use. The removable media path specifies which storage path is to be used by Windows to access System i tape and optical devices that have been allocated to Windows.

As with the default path for disk drives, you need to be careful that you do not overload a specific target HBA by assigning too many storage devices (storage spaces, optical drives, and tape drives) to the NWSH for the target HBA.

The removable media path defaults to the NWSH that you specify in the INSWNTSVR command when you create the Windows server.

9.7.3 Path deployment considerations

There are considerations you need to be aware of when planning for path deployment. In particular, there is a function called Multipath I/O (MPIO) that is referred to in some of the i5/OS documentation. However, MPIO is not available in the initial implementation of iSCSI on System i.

To find out the latest status of MPIO, visit the following Web site:

<https://www.ibm.com/systems/i/bladecenter/iscsi/>

Data path redundancy and failover

Data path redundancy and automatic failover are functions of Multipath I/O and are therefore not yet available.

As it currently stands, on the hosting partition side, if a target HBA fails, you would lose all the storage space and Virtual Ethernet LAN connections that were using that target HBA. Although the possibility of an HBA failure is very slight, you might wish to minimize data path sharing across target HBAs.

On the initiator side, if an initiator HBA port fails, the data paths are automatically reassigned to another port (if one is available). Exceptions are the boot path and data paths that have been deployed manually. Automatic reassignment, using the path deployment function, occurs after a shutdown/restart of the hosted server.

Data path aggregation

Data path aggregation, which is also a function of Multipath I/O, occurs when a storage space in the hosting i5/OS partition can be simultaneously accessed by more than one target HBA. This could potentially provide increased bandwidth for disk accesses to a storage space because it is linked to multiple storage paths.

As an alternative to data path aggregation, you could create multiple storage spaces, link each one to a different target HBA, then create a volume set in Windows over these storage spaces. This would provide a similar capability to the data path aggregation function of Multipath I/O in terms of distributing the workload across multiple target HBAs.

Multi-path groups

You see references to multi-path groups and multi-path group IQNs in some of the iSeries Navigator panels; for example, Figure 9-18 on page 366. Multi-path groups are also part of Multipath I/O and are therefore not yet available.

Meshed data flow

Multiple target HBAs (up to four) can establish connections to a single initiator HBA, but you currently cannot have a situation where a single target HBA is communicating with multiple initiator HBA ports on the same xSeries or Blade server. This meshed data flow, as shown in Figure 9-19, is currently *not* supported.

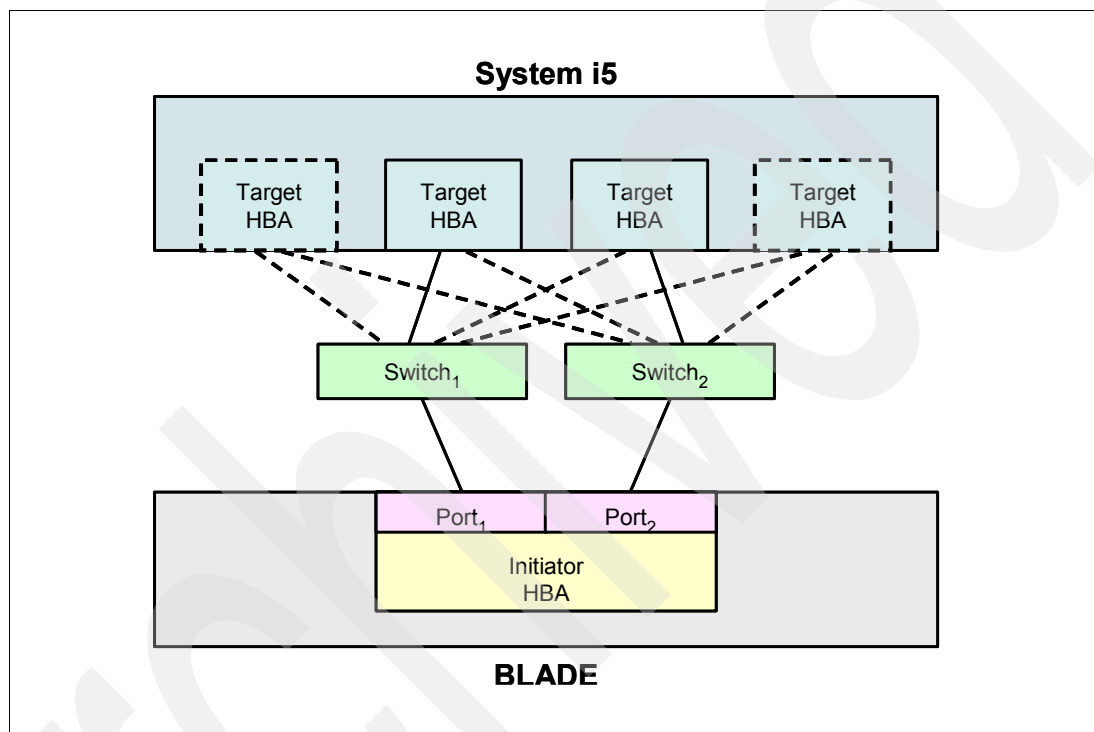


Figure 9-19 Meshed data flow is not supported

Showing this another way, the deployment scenario shown in Figure 9-20 on page 369 is valid, but the one shown in Figure 9-21 on page 369 is not.

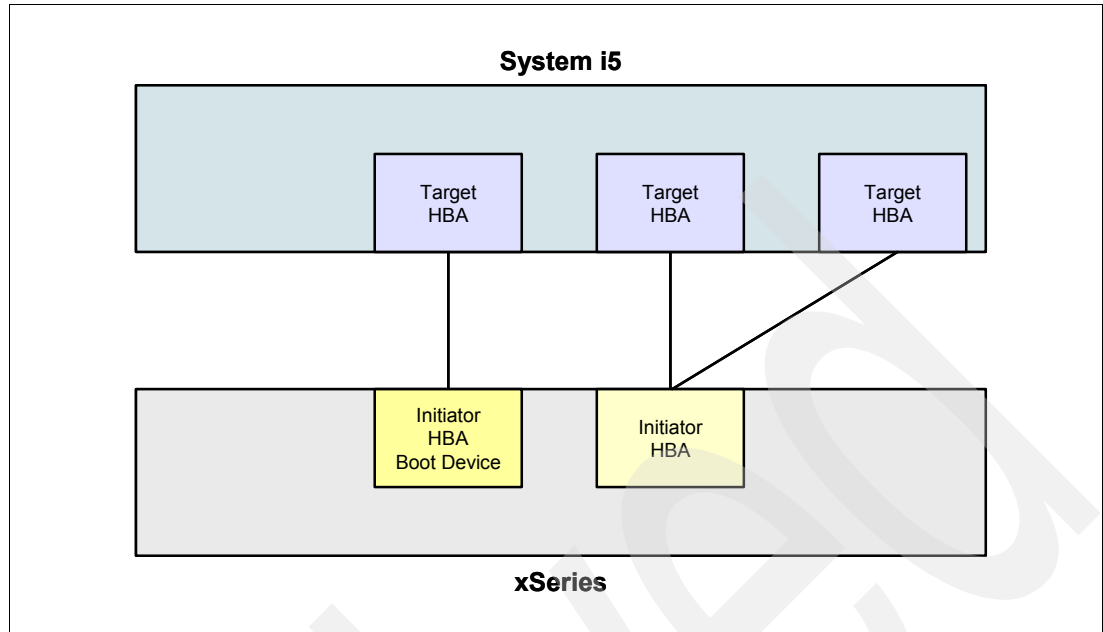


Figure 9-20 A single initiator HBA port can communicate with multiple target HBAs (up to four)

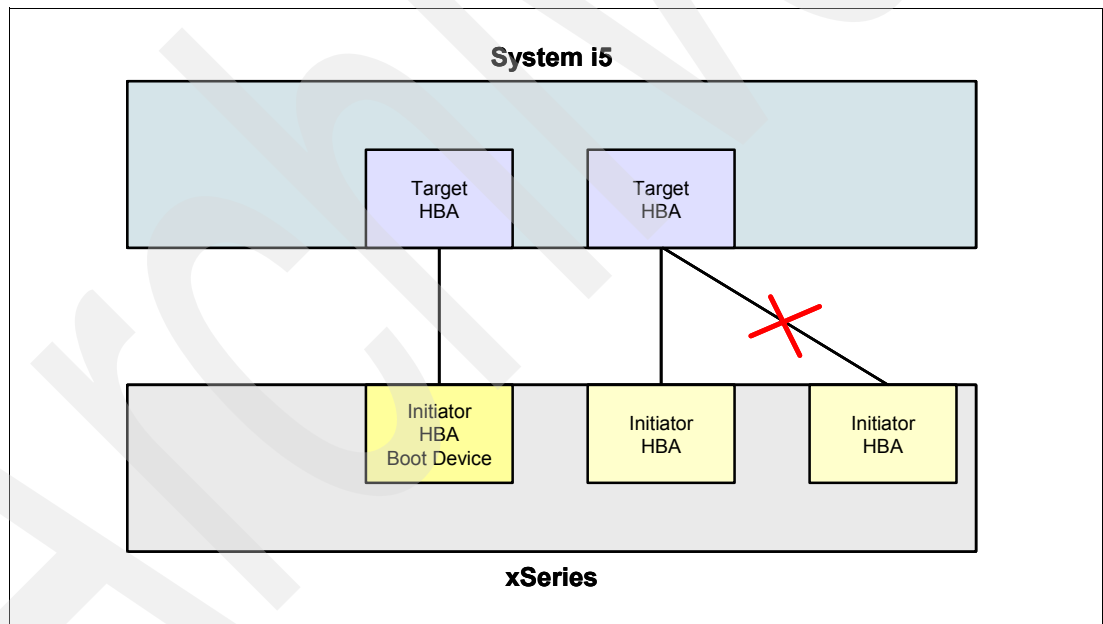


Figure 9-21 A single target HBA cannot communicate with multiple initiator HBA ports

9.8 Automatic path deployment

We cover the following topics in this section:

► Automatic path deployment - xSeries server

Here we look at automatic path deployment on a hosted xSeries server from a physical connectivity perspective, as well as how the data paths are established across the iSCSI network.

Refer to 9.8.1, “Automatic path deployment (physical view) - xSeries server” on page 370, and 9.8.2, “Automatic path deployment (data flow view) - xSeries server” on page 372.

► **Automatic path deployment - Blade server**

Here we look at automatic path deployment on a hosted Blade server from a physical connectivity perspective, as well as how the data paths are established across the iSCSI network.

Refer to 9.8.3, “Automatic path deployment (physical view) - Blade server” on page 374, and 9.8.4, “Automatic path deployment (data flow view) - Blade server” on page 375.

As previously mentioned, automatic path deployment occurs on the hosted server (initiator) side only, and only when you do not choose to manually select data paths. Autodeployment allocates initiator HBA ports to Virtual Ethernet LANs and non-boot virtual disks. IBM proprietary algorithms, running in Windows, select the best data path between target and initiator HBA ports to form a connection. The boot drives are a special case as described in “The boot path” on page 365.

If you install additional initiator HBA ports in an xSeries server, or activate the second initiator HBA port in a Blade server, data paths are automatically redeployed using the IBM proprietary algorithms. This redeployment occurs after the next shutdown/restart of the hosted server.

Automatic path deployment does not try to balance the flow of data through different initiator HBA ports, although it does try to spread the number of connections equally across the initiator HBA ports in the xSeries or Blade server. Therefore, if you want to try and balance data flows across initiator HBA ports, you need to consider deploying your data paths manually as described in 9.9, “Manual path deployment - target side” on page 377.

9.8.1 Automatic path deployment (physical view) - xSeries server

Figure 9-22 on page 371 shows how autodeployment works in a fully scaled xSeries server scenario.

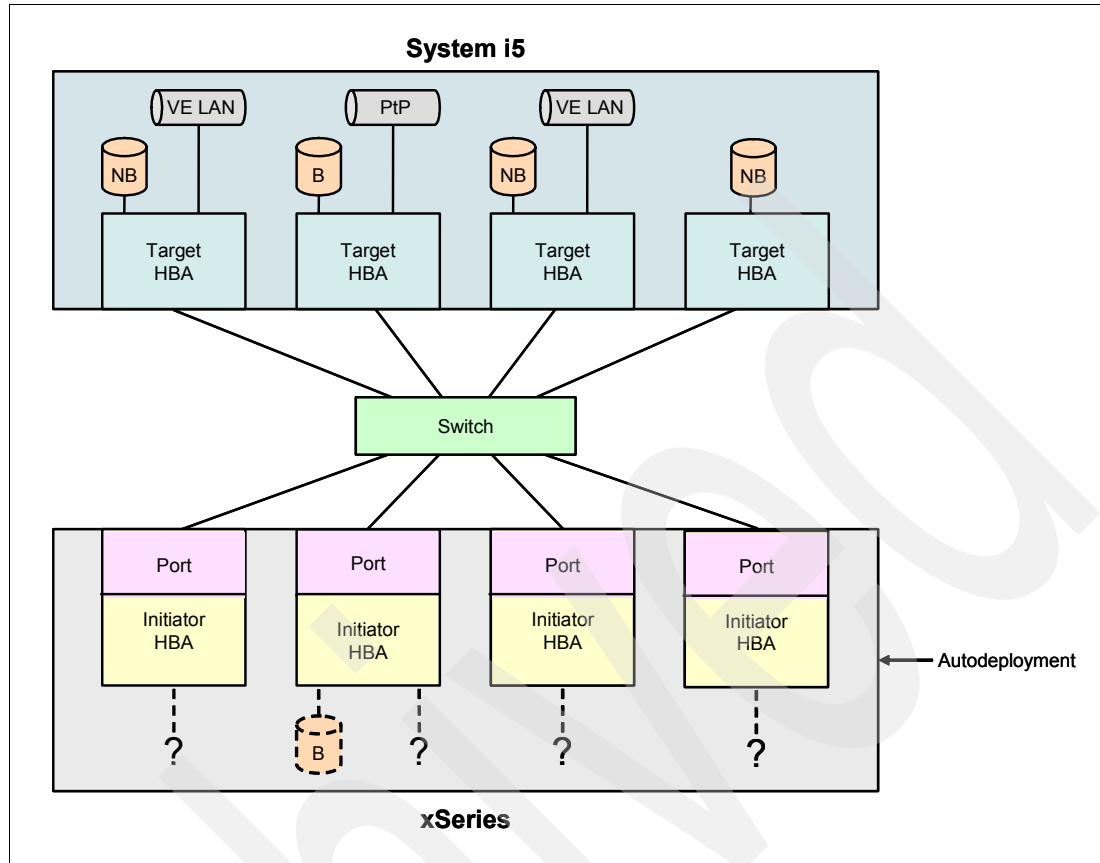


Figure 9-22 Autodeployment - xSeries

In Figure 9-22, we see that a hosted xSeries server has been installed as follows:

- ▶ Four target HBAs have been assigned to the NWSD for this hosted server using the four NWSH objects created for the targets (one per target). This provides up to four storage paths for the NWSD which is the maximum number. (Remember that the number of target HBAs communicating with the hosted server must be equal to or greater than the number of initiator HBAs installed in the server.)
- ▶ The xSeries server has the maximum number of initiator HBA ports installed, which is four.
- ▶ There are three storage spaces (NB) configured for the hosted server in addition to the C: and D: boot drives (B). The storage spaces have been linked to four different target HBAs. One of the initiator HBA ports has been nominated as the boot port in the RMTSYS object.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. Each of the Virtual Ethernet LANs is assigned to a different target HBA.
- ▶ The question marks signify that we do not know which initiator HBAs are used to establish connections with each non-boot drive and Virtual Ethernet LAN.

Conclusion

The non-boot SCSI and Virtual Ethernet LAN data paths are established automatically using autodeployment.

9.8.2 Automatic path deployment (data flow view) - xSeries server

We work through a couple of examples of data flow to help you understand how autodeployment works on xSeries servers.

Example 1: Basic iSCSI configuration

Figure 9-23 shows how data paths are deployed for a basic iSCSI configuration. Note that this example is the same for both automatic deployment and manual deployment on an xSeries server because there is only one target-initiator HBA port pair, and therefore only one data path.

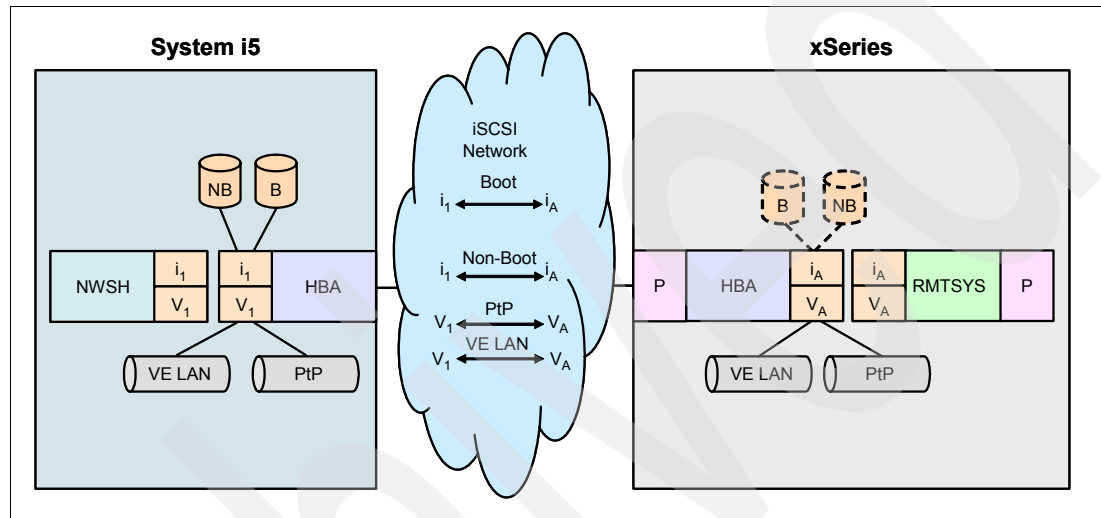


Figure 9-23 1:1:1:1:X

Read Figure 9-23 as follows:

- ▶ In the System i partition, there is a single target HBA with a SCSI IP address (i_1) and a Virtual Ethernet LAN IP address (V_1). i_1 and V_1 are stored in the NWSH object, which describes the target HBA.
- ▶ A non-boot drive (NB) has been configured for the hosted server and linked to the target HBA via a storage path.
- ▶ A Virtual Ethernet LAN has been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. It has been assigned to the target HBA.
- ▶ In the xSeries server, there is a single initiator HBA port with a SCSI IP address (i_A) and a Virtual Ethernet LAN IP address (V_A). i_A and V_A are stored in the RMTSYS object, which describes the hosted server.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the xSeries server to illustrate the concept.

Conclusion

Because there is only one initiator HBA port in this example, deployment of the SCSI and Virtual Ethernet LAN data paths occurs automatically across the iSCSI network. A single SCSI connection is established between i_1 and i_A , and a single LAN connection is established between V_1 and V_A across the iSCSI network.

Example 2: Scaled up iSCSI configuration

Figure 9-24 on page 373 shows how data paths are deployed automatically for a scaled up iSCSI configuration.

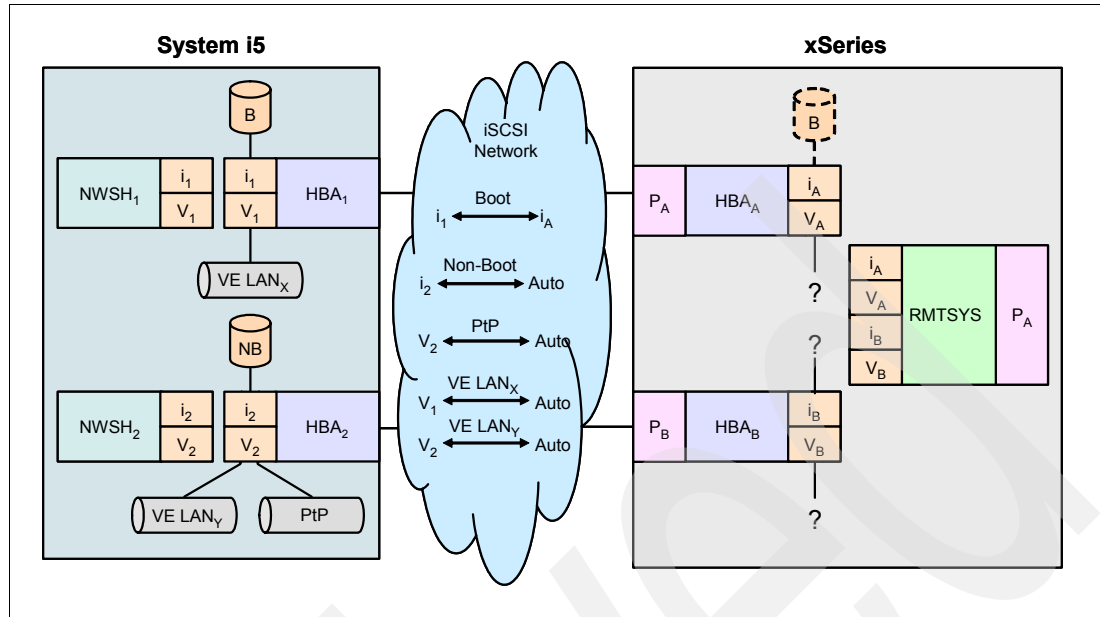


Figure 9-24 2 : 1 : 2 : 1 : X

Read Figure 9-24 as follows:

- ▶ In the System i partition, there are two target HBAs, each with a SCSI IP address (i₁ and i₂) and a Virtual Ethernet LAN IP address (V₁ and V₂). i₁ and V₁ are stored in the NWSH₁ object, which describes target HBA₁, and i₂ and V₂ are stored in the NWSH₂ object, which describes target HBA₂.
- ▶ The boot drives (B) are linked to NWSH₁ because this was the storage path specified in the INSWNTSVR command. Therefore, NWSH₁ is the default path.
- ▶ A non-boot drive (NB) has been configured for the hosted server and linked to target HBA₂ via a storage path.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. One has been assigned to target HBA₁ and the other to target HBA₂.
- ▶ In the xSeries server, there are two initiator HBA ports, each with a SCSI IP address (i_A and i_B) and a Virtual Ethernet LAN IP address (V_A and V_B). i_A, i_B, V_A, and V_B are stored in the RMTSYS object, which describes the hosted server.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the xSeries server to illustrate the concept.

- ▶ P_A has been specified as the boot port in the RMTSYS object. Therefore, a connection is established between i₁ and i_A for the boot path.
- ▶ The question marks signify that we do not know which initiator HBAs are used to establish connections with each non-boot drive and Virtual Ethernet LAN.

Conclusions

Because there are two initiator HBA ports in this example, automatic deployment comes into play as follows:

- ▶ A SCSI connection is established between i₁ and i_A for the boot drives.
- ▶ A SCSI connection is established between i₂ and either i_A or i_B for the non-boot drive. This connection is automatically deployed.

- ▶ A Virtual Ethernet LAN connection is established between V_1 and either V_A or V_B for Virtual Ethernet LAN_x across the iSCSI network. This connection is automatically deployed.
- ▶ A Virtual Ethernet LAN connection is established between V_2 and either V_A or V_B for both Virtual Ethernet LAN_y and the point-to-point Virtual Ethernet LAN across the iSCSI network. This connection is automatically deployed.

9.8.3 Automatic path deployment (physical view) - Blade server

Figure 9-25 shows how autodeployment works in a fully scaled Blade server scenario.

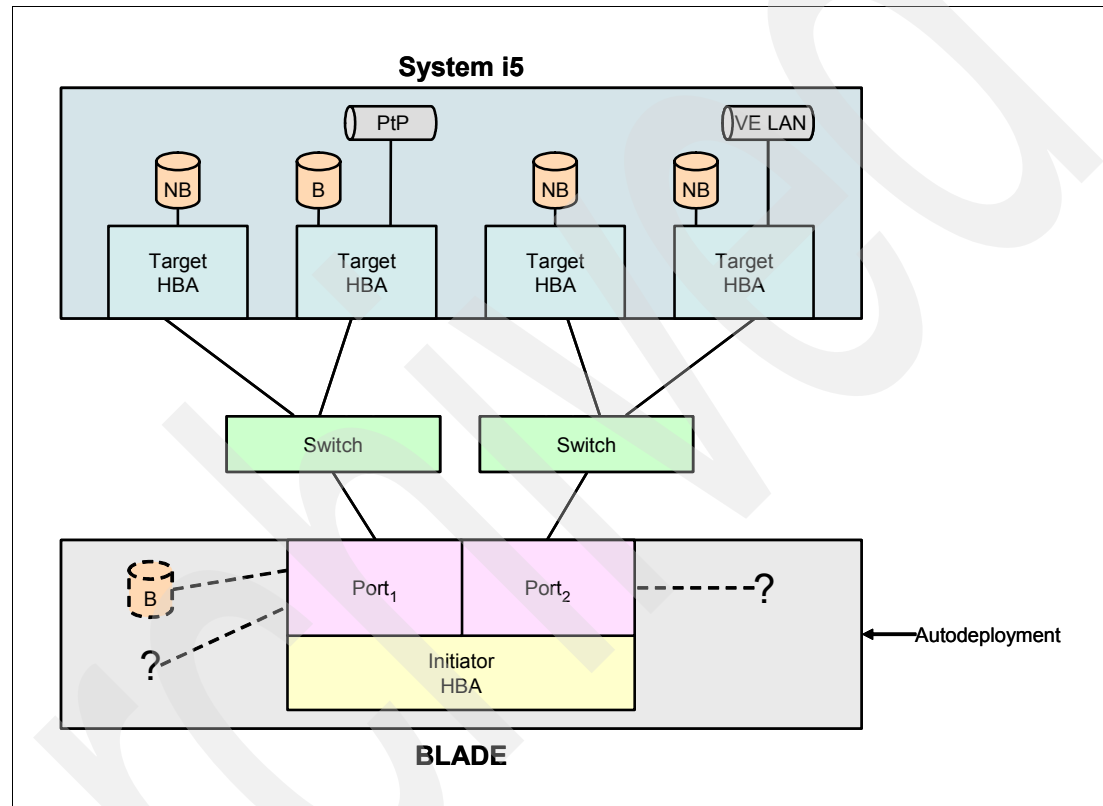


Figure 9-25 Autodeployment - Blade

In Figure 9-25, we see that a hosted Blade server has been installed as follows:

- ▶ Four target HBAs have been assigned to the NWSD for this hosted server using the four NWSH objects created for the targets (one per target). This provides up to four storage paths for the NWSD which is the maximum number. (Remember that the number of target HBAs communicating with the hosted server must be equal to or greater than the number of initiator HBAs installed in the server.)
- ▶ The Blade server has the maximum number of initiator HBA ports installed, which is two.
- ▶ There are three storage spaces (NB) configured for the hosted server in addition to the C: and D: boot drives (B). The storage spaces have been linked to four different target HBAs. Port 1 on the initiator HBA has been nominated as the boot port in the RMTSYS object.
- ▶ A Virtual Ethernet LAN has been configured to communicate with the hosted server in addition to the point-to-point Virtual Ethernet LAN. Each of the Virtual Ethernet LANs is assigned to a different target HBA.

- The question marks signify that we do not know which initiator HBA ports are used to establish connections with each non-boot drive and Virtual Ethernet LAN.

Conclusion

The non-boot SCSI and Virtual Ethernet LAN data paths are established automatically using autodeployment.

9.8.4 Automatic path deployment (data flow view) - Blade server

We work through a couple of examples of data flow to help you understand how autodeployment works on Blade servers.

Example 1: Basic iSCSI configuration

Figure 9-26 shows how data paths are deployed for a basic iSCSI configuration. Note that this example is the same for both automatic deployment and manual deployment on a Blade server because there is only one target-initiator HBA port pair, and therefore only one data path.

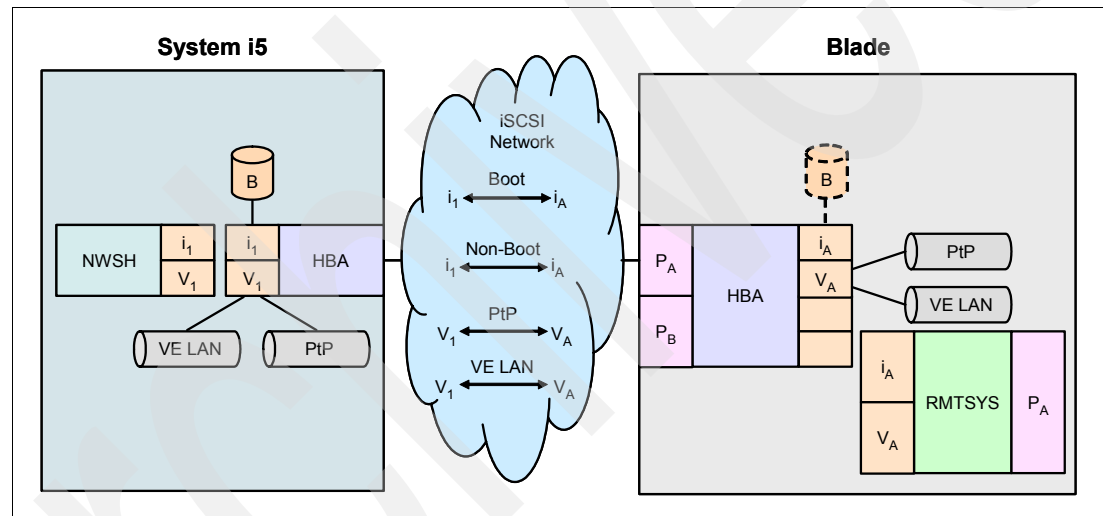


Figure 9-26 1 : 1 : 1 : 1 : B

Read Figure 9-26 as follows:

- In the System i partition, there is a single target HBA with a SCSI IP address (i_1) and a Virtual Ethernet LAN IP address (V_1). i_1 and V_1 are stored in the NWSH object, which describes the target HBA.
- A Virtual Ethernet LAN has been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. It has been assigned to the target HBA.
- In the Blade server, there is a single initiator HBA port with a SCSI IP address (i_A) and a Virtual Ethernet LAN IP address (V_A). i_A and V_A are stored in the RMTSYS object, which describes the hosted server. The second port on the initiator HBA is not activated, and therefore the IP addresses are blank.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the Blade server to illustrate the concept.

Conclusion

Because there is only one initiator HBA port in this example, deployment of the SCSI and Virtual Ethernet LAN data paths occurs automatically across the iSCSI network. A single

SCSI connection is established between i_1 and i_A , and a single LAN connection is established between V_1 and V_A across the iSCSI network.

Example 2: Scaled up iSCSI configuration

Figure 9-27 shows how data paths are deployed automatically for a scaled up iSCSI configuration.

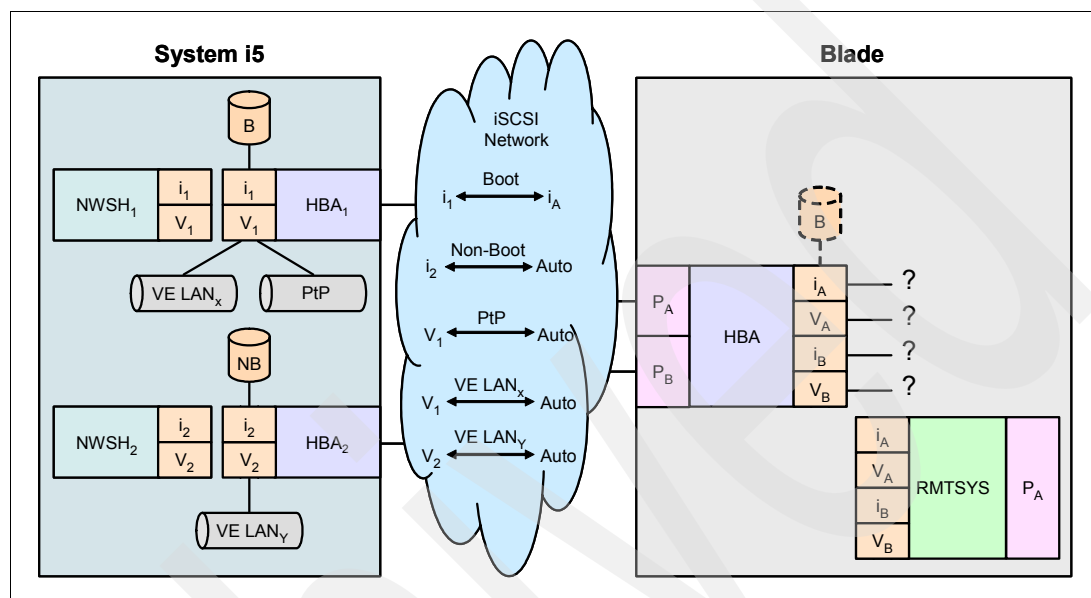


Figure 9-27 2 : 1 : 2 : 1 : B

Read Figure 9-27 as follows:

- ▶ In the System i partition, there are two target HBAs, each with a SCSI IP address (i_1 and i_2) and a Virtual Ethernet LAN IP address (V_1 and V_2). i_1 and V_1 are stored in the NWSH₁ object, which describes target HBA₁, and i_2 and V_2 are stored in the NWSH₂ object, which describes target HBA₂.
- ▶ The boot drives (B) are linked to NWSH₁ because this was the storage path specified in the INSWNTSVR command. Therefore, NWSH₁ is the default path.
- ▶ An additional non-boot drive (NB) has been configured for the hosted server and linked to target HBA₂ via a storage path.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. One has been assigned to target HBA₁ and the other to target HBA₂.
- ▶ In the Blade server, there are two initiator HBA ports, each with a SCSI IP address (i_A and i_B) and a Virtual Ethernet LAN IP address (V_A and V_B). i_A , i_B , V_A , and V_B are stored in the RMTSYS object, which describes the hosted server.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the xSeries server to illustrate the concept.

- ▶ P_A has been specified as the boot port in the RMTSYS object. Therefore, a connection is established between i_1 and i_A for the boot path.
- ▶ The question marks signify that we do not know which initiator HBAs are used to establish connections with each non-boot drive and Virtual Ethernet LAN.

Conclusions

Because there are two initiator HBA ports in this example, automatic deployment comes into play as follows:

- ▶ A SCSI connection is established between i_1 and i_A for the boot drives.
- ▶ A SCSI connection is established between i_2 and either i_A or i_B for the non-boot drive. This connection is automatically deployed.
- ▶ A Virtual Ethernet LAN connection is established between V_1 and either V_A or V_B for Virtual Ethernet LAN_x and the point-to-point Virtual Ethernet LAN across the iSCSI network. This connection is automatically deployed.
- ▶ A Virtual Ethernet LAN connection is established between V_2 and either V_A or V_B for Virtual Ethernet LAN_y across the iSCSI network. This connection is automatically deployed.

9.9 Manual path deployment - target side

Manual path deployment, or *redemption*, can occur on both the target side and the initiator side and enables you to specify which target or initiator HBA port you want to use for both storage space links and Virtual Ethernet LAN connections.

Storage space links and Virtual Ethernet LAN connections are always deployed manually on the target side. That is, you always need to specifically link a storage space, or assign a Virtual Ethernet LAN, to a target HBA in order to use the storage space or Virtual Ethernet LAN. The only exceptions are the C: and D: drives, which are linked automatically, and the point-to-point Virtual Ethernet LAN, which is assigned automatically when the Windows server is created.

Data paths are manually deployed on the target side for both storage spaces and Virtual Ethernet LANs as follows:

▶ Deploying storage spaces on the target side

On the target side of the iSCSI network, you can link any storage space to any target HBA as long as the NWSH object for the target HBA is configured in the NWSD as a storage path. The NWSD exposes the target HBA (NWSH) and its linked storage spaces to the hosted server via the RMTSYS configuration object. There is always a 1 : 1 relationship between the NWSD and RMTSYS for a hosted server. Note that the C: and D: drive storage spaces (boot drives) must always be linked to the same NWSH.

▶ Deploying Virtual Ethernet LANs on the target side

On the target side of the iSCSI network, you can assign any Virtual Ethernet LAN to any target HBA by specifying the NWSH object for the target HBA in the NWSD. The NWSD exposes the target HBA (NWSH) and its assigned Virtual Ethernet LANs to the hosted server via the RMTSYS configuration object. There is always a 1 : 1 relationship between the NWSD and RMTSYS for a hosted server.

Figure 9-28 on page 378 shows how manual deployment works on the target side in a fully scaled xSeries server scenario.

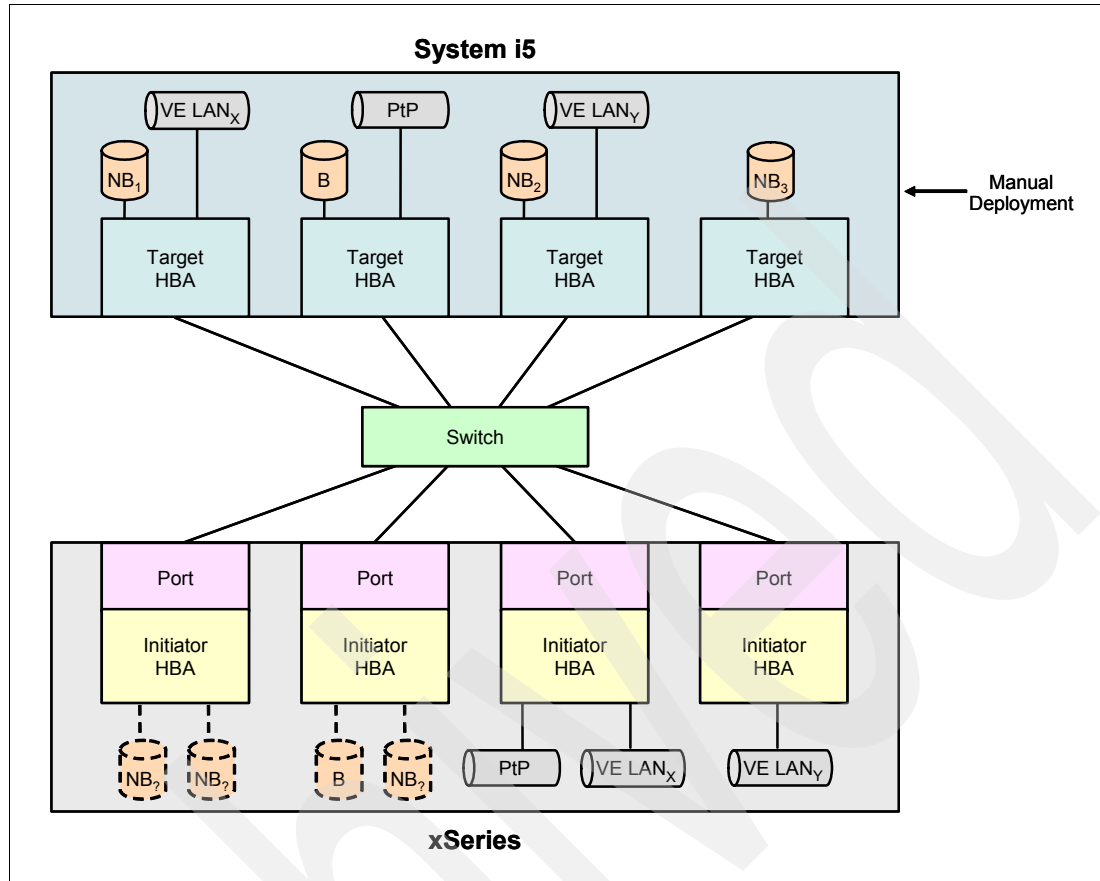


Figure 9-28 Manual deployment on the target side - xSeries

Manual deployment on the target side occurs the same way for both an xSeries and Blade server. Therefore, we only show the xSeries server example.

In Figure 9-28, we see that a hosted xSeries server has been installed as follows:

- ▶ Four target HBAs have been assigned to the NWSD for this hosted server via four NWSH configuration objects. Four target HBAs are the maximum for a hosted server. (Remember that the number of target HBAs communicating with a hosted server must be equal to or greater than the number of initiator HBAs installed in the server.)
- ▶ There are three storage spaces configured for the hosted server in addition to the boot drives (C: and D:). All the storage spaces have been linked to different target HBAs via their respective storage paths.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server in addition to the point-to-point Virtual Ethernet LAN. All the Virtual Ethernet LANs have been assigned to different target HBAs.

Conclusion

We manually deploy each storage space and Virtual Ethernet LAN by specifying which target HBA for it to use.

Setup

In this chapter, we assume that all storage spaces and Virtual Ethernet LANs have been previously created for the hosted server. In this case, as we add target HBAs, we need to *redeploy* the existing storage spaces and Virtual Ethernet LANs from one target HBA to

another to spread the workload evenly and achieve optimal performance. Therefore, we describe how to *redeploy* existing storage space links and Virtual Ethernet connections, rather than how to deploy new ones.

To manually redeploy an existing storage space link or Virtual Ethernet LAN connection on the target side, you need to complete one of the following tasks:

- ▶ **Redeploy a storage space link to a different target HBA**
Refer to 9.15.1, “Redeploying a storage space to a different target HBA” on page 437.
- ▶ **Redeploy a Virtual Ethernet LAN connection to a different target HBA**
Refer to 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.

9.10 Manual path deployment - initiator side

We cover the following topics in this section:

- ▶ **Overview**
Here we provide an introduction to how SCSI and Virtual Ethernet LAN connections can be tied to a specific initiator HBA port.
Refer to 9.10.1, “Overview” on page 379.
- ▶ **Manual path deployment - xSeries**
Here we look at manual path deployment on a hosted xSeries server from a physical connectivity perspective, as well as how the data paths are established across the iSCSI network.
Refer to 9.10.2, “Manually deploying data paths on the initiator side - xSeries server” on page 380 and 9.10.3, “Manual path deployment (data flow view) - xSeries server” on page 382.
- ▶ **Manual path deployment - Blade**
Here we look at manual path deployment on a hosted Blade server from a physical connectivity perspective, as well as how the data paths are established across the iSCSI network.
Refer to 9.10.4, “Manually deploying data paths on the initiator side - Blade server” on page 384 and 9.10.5, “Manual path deployment (data flow view) - Blade server” on page 385.

9.10.1 Overview

Manual path deployment, or redeployment, can occur on both the target side and the initiator side and enables you to specify which target or initiator HBA port you want to use for both storage space links and Virtual Ethernet LAN connections.

On the initiator side, storage space links and Virtual Ethernet LAN connections can be automatically deployed, manually deployed, or a combination of both. If you do not manually assign a non-boot disk drive or Virtual Ethernet LAN to an initiator HBA port, then the data path is deployed automatically as described in 9.8, “Automatic path deployment” on page 369.

Data paths can be manually deployed on the initiator side for both virtual disks and Virtual Ethernet LANs as follows:

► **Deploying virtual disks on the initiator side**

On the initiator side, you must select which initiator HBA port the Windows server boots from (assuming that there are two or more ports). This is called the boot port, and together with the target HBA that the boot drives are linked to, establishes the boot path. The boot path is a special case because it must always be manually deployed. For more information about the boot path, refer to “The boot path” on page 365.

Non-boot drives can be manually deployed at the initiator end. This is done by blanking out the SCSI IP address of the SCSI-LAN IP address pair specified in the RMTSYS configuration object. While this might seem crude, it is the only way to manually deploy non-boot drives on the initiator side. It has the effect of selecting those initiator HBA ports that you do *not* want virtual disks to use rather than those you *do* want them to use.

The steps you need to perform to manually deploy a virtual disk on the initiator side are described in 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.

► **Deploying Virtual Ethernet LANs on the initiator side**

Virtual Ethernet LAN connections, including the point-to-point Virtual Ethernet LAN, appear as icons in Windows Network Connections. You can switch a Virtual Ethernet LAN from one initiator HBA port to another by reconfiguring its Windows network connection.

The steps you need to perform to manually deploy a Virtual Ethernet LAN connection on the initiator side are described in 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

Note that you can also manually deploy Virtual Ethernet LAN connections on the initiator side by blanking out the LAN IP address of the SCSI-LAN IP address pair specified in the RMTSYS configuration object (as you can for non-boot virtual disks). However, we do not cover this method because you can manually deploy Virtual Ethernet LAN connections more elegantly by changing the Properties of the corresponding Windows network connection.

Manually deploying data paths on the initiator side is slightly different for xSeries compared with Blade. We cover both cases here.

9.10.2 Manually deploying data paths on the initiator side - xSeries server

Figure 9-29 on page 381 shows how manual deployment works on the initiator side in a fully scaled xSeries server scenario.

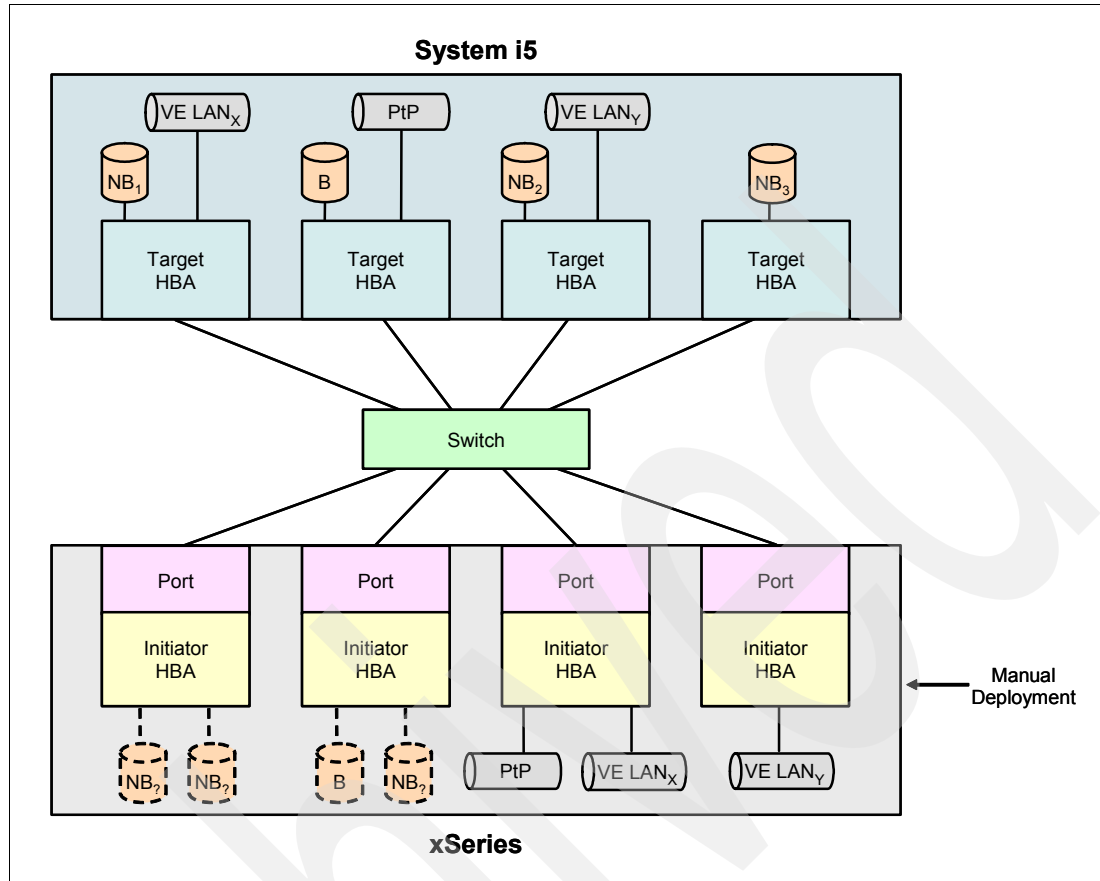


Figure 9-29 Manual deployment on the initiator side - xSeries

In Figure 9-29, we see that a hosted xSeries server has been installed as follows:

- ▶ Four initiator HBAs have been installed in the xSeries server, which is the maximum.
- ▶ There are three storage spaces configured for the hosted server in addition to the boot drives (C: and D:). One of the initiator HBA ports has been nominated as the boot port in the RMTSYS object.
- ▶ Two of the initiator HBA ports have been disabled for the deployment of SCSI connections by blanking out the SCSI Interface address information in the Network interface of the RMTSYS configuration object. Obviously, the boot port must always remain enabled. This has the effect of forcing the virtual disks to be automatically deployed to the two initiator HBA ports that have not been disabled. Because we have no control over which initiator HBA port the non-boot drives are deployed to, we denote these drives with question marks. Note that the path deployment algorithms attempt to evenly split the virtual disks across the available initiator HBA ports.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server in addition to the point-to-point Virtual Ethernet LAN. They are assigned to different initiator HBA ports by changing the IP address of their connections in Windows Network Connections.

Conclusions

We can draw the following conclusions from this example:

- ▶ Virtual disks can be manually deployed to a certain extent, but not at a virtual disk level. We can specify which initiator HBA ports *cannot* be used for deploying virtual disks, but we are not able to manually deploy a particular virtual disk to a specific initiator HBA port.
- ▶ Virtual Ethernet LANs *can* be manually deployed at a Virtual Ethernet LAN level to a specific initiator HBA port.

Setup

In this chapter, we assume that all virtual disks and Virtual Ethernet LANs have been previously created for the hosted server. In this case, as we add initiator HBA ports, we might need to *redeploy* the existing virtual disks and Virtual Ethernet LANs from one initiator HBA port to another to spread the workload evenly and achieve optimal performance. Therefore, we describe how to *redeploy* existing virtual disk assignments and Virtual Ethernet LAN connections, rather than how to deploy new ones.

To manually redeploy an existing virtual disk assignment or Virtual Ethernet LAN connection on the initiator side, you need to complete one or more of the following tasks, depending on your requirements:

- ▶ **Redeploy the boot path to a different initiator HBA port**
Refer to 9.15.3, “Changing the boot port” on page 443.
- ▶ **Redeploy a non-boot virtual disk to a different initiator HBA port**
Refer to 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
- ▶ **Redeploy a Virtual Ethernet LAN to a different initiator HBA port**
Refer to 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

9.10.3 Manual path deployment (data flow view) - xSeries server

We work through an example of data flow to help you understand how manual deployment works on xSeries servers.

Example 1: Basic iSCSI configuration

The data flow for this configuration is the same as for automatic deployment on an xSeries server because there is only one target-initiator HBA port pair. Therefore, we do not show it again. Refer to 9.8.2, “Automatic path deployment (data flow view) - xSeries server” on page 372 for a description.

Example 2: Scaled up iSCSI configuration

Figure 9-30 on page 383 shows how data paths are deployed manually for a scaled up iSCSI configuration.

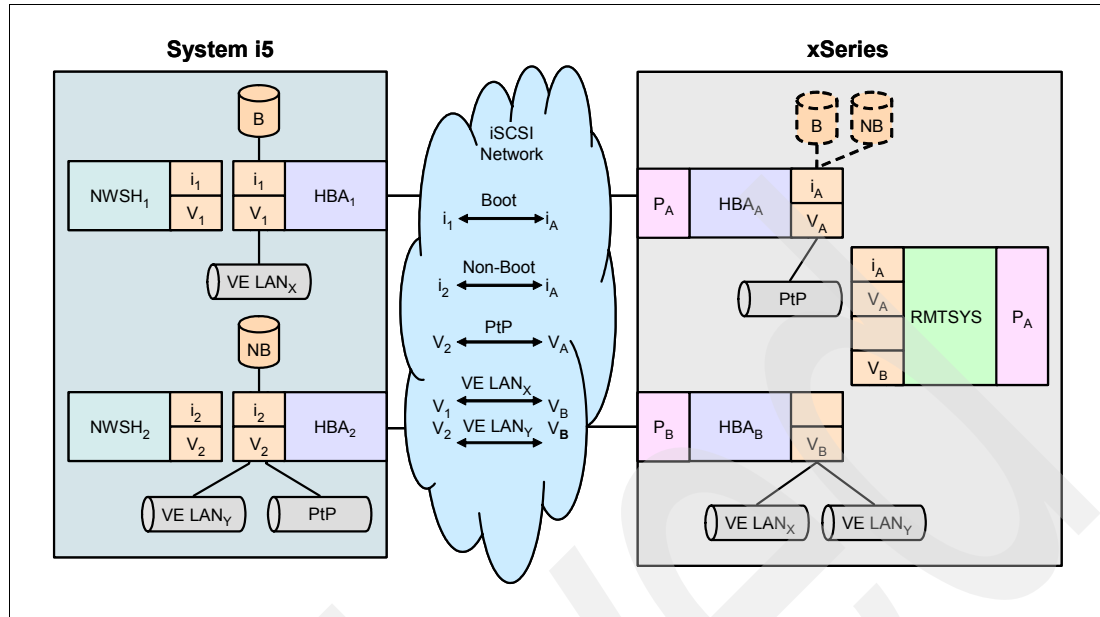


Figure 9-30 2:1:2:1:X

Read Figure 9-30 as follows:

- In the System i partition, there are two target HBAs, each with a SCSI IP address (i₁ and i₂) and a Virtual Ethernet LAN IP address (V₁ and V₂). i₁ and V₁ are stored in the NWSH₁ object, which describes target HBA₁, and i₂ and V₂ are stored in the NWSH₂ object, which describes target HBA₂.
- A non-boot drive (NB) has been configured for the hosted server in addition to the boot drives (C: and D:). The storage spaces have been linked to different target HBAs via their respective storage paths.
- Two Virtual Ethernet LANs have been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. They are assigned to different target HBAs.
- In the xSeries server, there are two initiator HBA ports, each with a SCSI IP address (i_A and i_B) and a Virtual Ethernet LAN IP address (V_A and V_B). i_A, i_B, V_A, and V_B are stored in the RMTSYS object, which describes the hosted server. Note that in this example, i_B has been blanked out, forcing the non-boot drive to be deployed to HBA_A.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the xSeries server to illustrate the concept.

- P_A has been specified as the boot port in the RMTSYS object. Therefore, a connection is established between i₁ and i_A for the boot path.

Conclusions

Because there are two initiator HBA ports in this example, virtual disks and Virtual Ethernet LANs can be manually deployed as follows:

- Initiator HBA port B has been disabled for the deployment of SCSI connections by blanking out the i_B SCSI Interface address information in the Network interface of the RMTSYS configuration object. Therefore, all virtual disks are deployed on port A.
- A Virtual Ethernet LAN connection is established between V₁ and V_B, and between V₂ and V_B because VE LAN_x and VE LAN_y were manually deployed to initiator HBA port B. Note that the point-to-point Virtual Ethernet LAN can also be manually deployed, but in this

example, it was not changed from the port it was deployed to when the hosted server was installed.

9.10.4 Manually deploying data paths on the initiator side - Blade server

Figure 9-31 shows how manual deployment works on the initiator side in a fully scaled Blade server scenario. There are a few differences compared to an xSeries server.

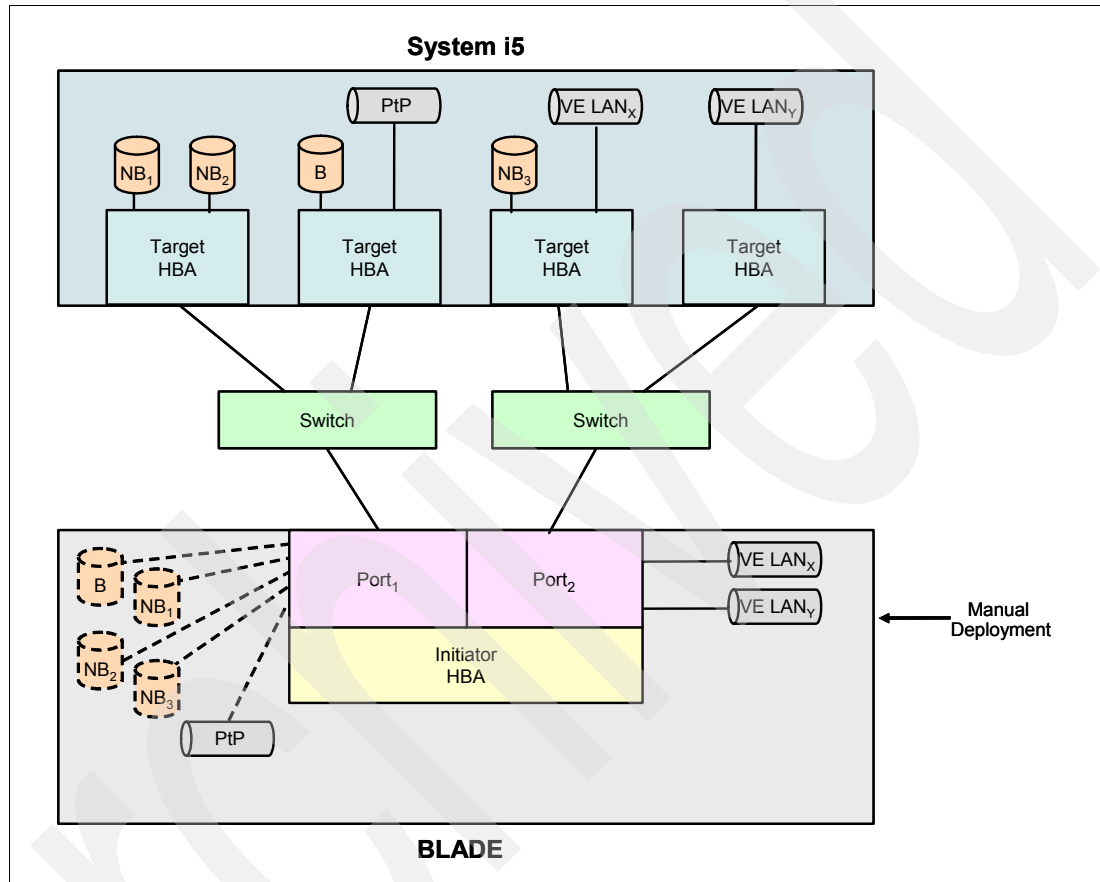


Figure 9-31 Manual deployment on the initiator side - Blade

In Figure 9-31, we see that a hosted Blade server has been installed as follows:

- ▶ Both initiator HBA ports have been activated in the Blade server, which is the maximum. Each initiator HBA port in the Blade server must be connected to a different switch module in the BladeCenter.
- ▶ There are three storage spaces configured for the hosted server in addition to the boot drives (C: and D:). One of the initiator HBA ports has been nominated as the boot port in the RMTSYS object.
- ▶ One of the initiator HBA ports has been disabled for the deployment of SCSI connections by blanking out the SCSI Interface address information in the Network interface of the RMTSYS configuration object. Obviously, the boot port must always remain enabled.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server in addition to the point-to-point Virtual Ethernet LAN. They are assigned to one of the initiator HBA ports by changing the IP address of the connection in Windows Network connections.

Conclusions

The conclusions are the same as for the xSeries server case, but note that, whereas you can manually deploy Virtual Ethernet LANs to either initiator HBA port, you can only manually deploy virtual disks to one port or the other. This is because you manually deploy virtual disks on the initiator side by disabling ports for the transmission of SCSI I/Os, and you cannot disable both ports on a Blade server's initiator HBA.

Setup

In this chapter, we assume that all virtual disks and Virtual Ethernet LANs have been previously created for the hosted server. In this case, if we add an initiator HBA port, we might need to *redeploy* the existing virtual disks and Virtual Ethernet LANs from one initiator HBA port to the other to spread the workload evenly and achieve optimal performance. Therefore, we describe how to *redeploy* existing virtual disk assignments and Virtual Ethernet LAN connections, rather than how to deploy new ones.

To manually redeploy an existing virtual disk assignment or Virtual Ethernet LAN connection on the initiator side, you need to complete one or more of the following tasks, depending on your requirements:

- ▶ **Redeploy the boot path to a different initiator HBA port**

Refer to 9.15.3, "Changing the boot port" on page 443.

- ▶ **Redeploy a non-boot virtual disk to a different initiator HBA port**

Refer to 9.15.4, "Preventing a non-boot drive from using an initiator HBA port" on page 444.

- ▶ **Redeploy a Virtual Ethernet LAN to a different initiator HBA port**

Refer to 9.15.5, "Redeploying a Virtual Ethernet LAN to a different initiator HBA port" on page 446.

9.10.5 Manual path deployment (data flow view) - Blade server

We work through an example of data flow to help you understand how manual deployment works on Blade servers.

Example 1: Basic iSCSI configuration

The data flow for this configuration is the same as for automatic deployment on a Blade server because there is only one target-initiator HBA port pair. Therefore, we do not show it again. Refer to 9.8.4, "Automatic path deployment (data flow view) - Blade server" on page 375 for a description.

Example 2: Scaled up iSCSI configuration

Figure 9-32 on page 386 shows how data paths are deployed manually for a scaled up iSCSI configuration.

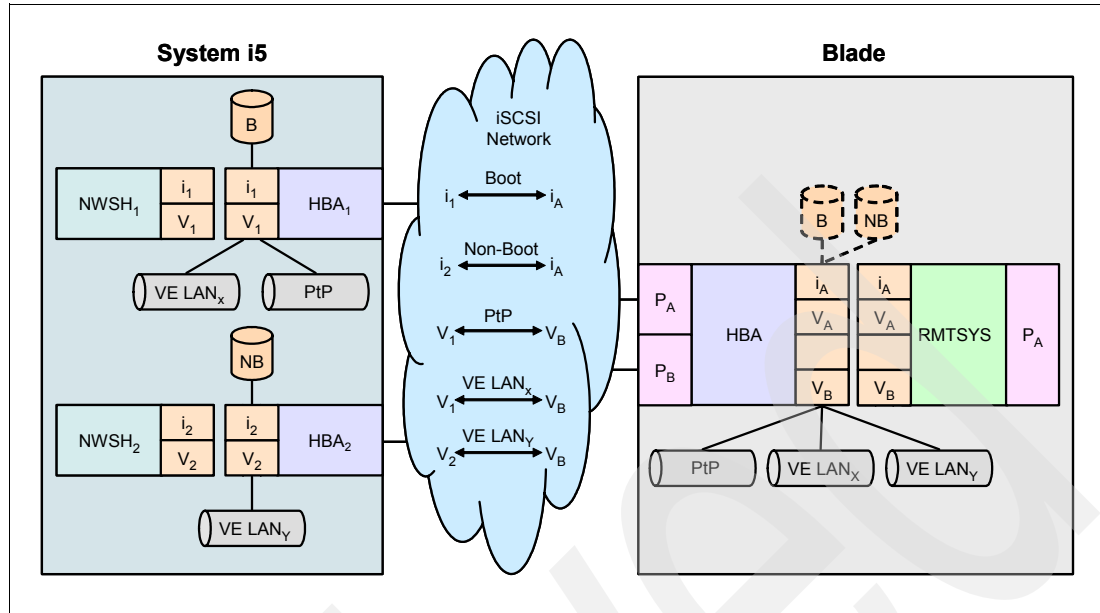


Figure 9-32 2 : 1 : 2 : 1 : B

Read Figure 9-32 as follows:

- ▶ In the System i partition, there are two target HBAs, each with a SCSI IP address (i₁ and i₂) and a Virtual Ethernet LAN IP address (V₁ and V₂). i₁ and V₁ are stored in the NWSH₁ object, which describes target HBA₁, and i₂ and V₂ are stored in the NWSH₂ object, which describes target HBA₂.
- ▶ A non-boot drive (NB) has been configured for the hosted server in addition to the boot drives (C: and D:). The storage spaces have been linked to different target HBAs via their respective storage paths.
- ▶ Two Virtual Ethernet LANs have been configured to communicate with the hosted server, in addition to the point-to-point Virtual Ethernet LAN. They are assigned to different target HBAs.
- ▶ In the Blade server, there are two initiator HBA ports, each with a SCSI IP address (i_A and i_B) and a Virtual Ethernet LAN IP address (V_A and V_B). i_A, i_B, V_A, and V_B are stored in the RMTSYS object, which describes the hosted server. Note that in this example, i_B has been blanked out, forcing the non-boot drive to be deployed to P_A.

Note that the RMTSYS object is physically stored in the hosting partition but is drawn inside the Blade server to illustrate the concept.

- ▶ P_A has been specified as the boot port in the RMTSYS object. Therefore, a connection is established between i₁ and i_A for the boot path.

Conclusions

Because there are two initiator HBA ports in this example, virtual disks and Virtual Ethernet LANs can be manually deployed as follows:

- ▶ Initiator HBA port B has been disabled for the deployment of SCSI connections by blanking out the i_B SCSI Interface address information in the Network interface of the RMTSYS configuration object. Therefore, all virtual disks are deployed on port A.
- ▶ A Virtual Ethernet LAN connection is established between V₁ and V_B, and between V₂ and V_B because VE LAN_x and VE LAN_y were manually deployed to initiator HBA port B. Note that the point-to-point Virtual Ethernet LAN has also been manually deployed to port B.

9.11 Capacity planning for iSCSI

In this section, we discuss the basic rules of capacity planning as they relate to the implementation of iSCSI on the System i.

The most important time that you need to perform capacity planning is when you are initially sizing the System i iSCSI network. However, capacity planning should also be performed when you are planning to connect additional hosted servers, or when you want to increase the bandwidth of a specific hosted server connection. In each case, you need to be able to calculate the following System i resource requirements to accurately determine the hardware configuration needed to support your iSCSI network:

- ▶ The number of target HBAs
- ▶ Processor CPW
- ▶ Memory
- ▶ Disk storage

Accurate capacity planning will ensure that your System i iSCSI network performs optimally.

We cover the following topics:

- ▶ **Introduction**

Here we introduce the topics of performance and capacity planning as they relate to iSCSI on System i, and describe possible test environments.

Refer to 9.11.1, “Introduction” on page 387.

- ▶ **Getting started with capacity planning for iSCSI**

Here we list the steps you need to work through to properly plan for your System i iSCSI network.

Refer to 9.11.2, “Getting started with capacity planning for iSCSI” on page 391.

- ▶ **Windows Performance Monitor**

Here we describe how you can collect performance data by using the Windows Performance Monitor.

Refer to 9.11.3, “Obtain Windows performance data” on page 392.

- ▶ **Capacity planning general rules**

Here we present basic capacity planning rules that will help you to plan the hardware required for your System i iSCSI network.

Refer to 9.11, “Capacity planning for iSCSI” on page 387.

- ▶ **Scalability limits**

Here we document the various scalability limits of the System i iSCSI network.

Refer to 9.11.8, “Scalability limits” on page 399.

9.11.1 Introduction

The new iSCSI solution on System i offers greater scalability and improved overall performance compared to the IXS/IXA technology. iSCSI also has the advantage of supporting IBM BladeCenter, and being more flexible in its implementation in terms of the number of Windows servers that can be supported. However, compared with IXS/IXA, the resource requirements of the new iSCSI implementation have a slightly higher System i CPU cost as well as additional memory overhead.

Some preliminary work has been done to compare the performance of the new iSCSI HBAs against the HSL-based IXS/IXA technology. Figure 9-33 shows the throughput in MB/sec of a single target HBA communicating with a single initiator HBA port in a Blade server using various disk operation sizes. Note that the disk operation size is the amount of data in kilobytes that a Windows application requests to be read from, or written to disk in a single disk access.

The throughput of the target HBA in Figure 9-33 (and also generally) depends on a number of factors, which include:

- ▶ The throughput capacity of the target HBA itself
- ▶ The throughput capacity of the iSCSI network:
 - This is controlled by the throughput and latency of the network switches.
- ▶ The System i configuration:
 - The disk configuration:
 - The number of disk arms
 - The type of disk controller
 - Whether the disks are configured for RAID or mirroring
 - The amount of processing power available
 - The amount of memory configured

Therefore, the throughput values shown in Figure 9-33 might not necessarily reflect the maximum throughput of a target HBA in an unconstrained environment. (In this case, an unconstrained environment means that the target HBA is the bottleneck; no other resources are limiting its throughput.) What Figure 9-33 does allow us to do is compare the throughput of iSCSI against the throughput of IXA for the same System i and System x™ hardware configuration.

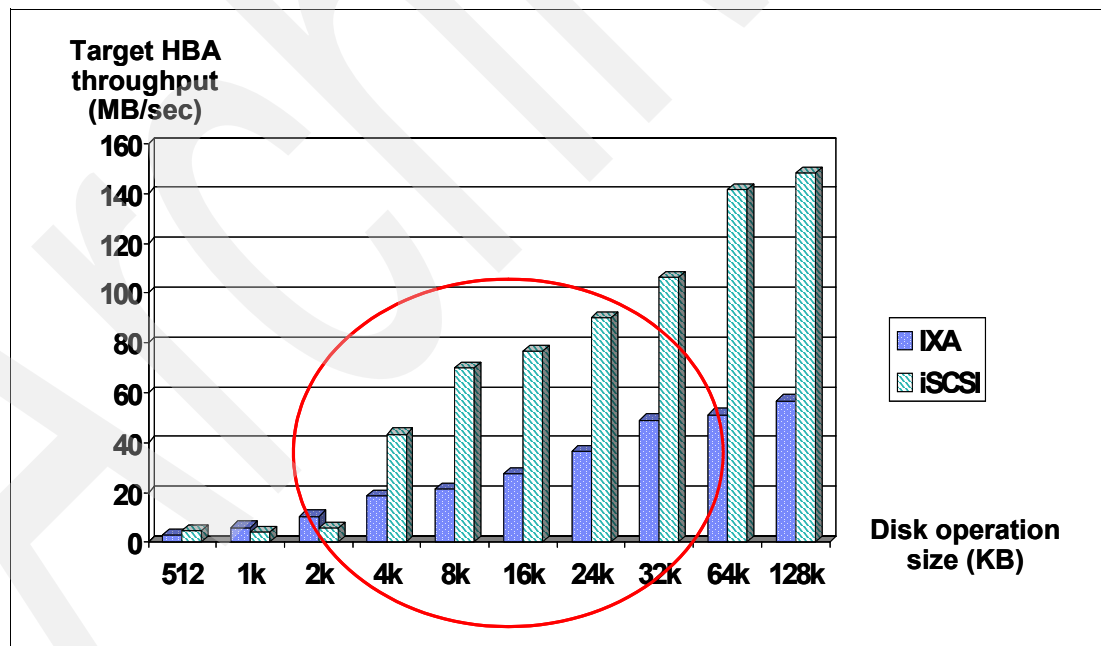


Figure 9-33 Performance of iSCSI versus IXA for a single target-initiator HBA port pair

Figure 9-33 shows that for disk operation sizes of 4 KB and above, the throughput of an iSCSI target-initiator HBA port pair is at least twice the throughput of an HSL attached IXA. Below 4 KB, the performance of both iSCSI and HSL is substantially reduced. This demonstrates that the new page-based I/O architecture utilized in the iSCSI implementation provides better overall performance in read/write scenarios, while the sector-based I/O architecture of the

IXA technology provides slightly better performance for small block write operations. 4 K - 24 K disk operation sizes are generally more representative of Windows I/O workloads.

Given an “unconstrained” environment, where the target HBA is the bottleneck, the maximum throughput of a single target HBA has been estimated at approximately 90 MB/sec, based on an 8 K disk operation size. However, as previously mentioned, the maximum throughput for a particular hardware configuration will depend on a number of factors that can reduce the throughput to less than the 90 MB/sec value. For example, Figure 9-33 on page 388 shows that the measured throughput of a target HBA using an 8 KB disk operation size in this particular test environment was in fact less than 80 MB/sec. This indicates that there was something in the test configuration, other than the target HBA, that was limiting the throughput.

The size of the disk operations is a function of the application running on the Windows server. As shown by the ellipse in Figure 9-33 on page 388, most Windows applications use disk operation sizes between 4 KB and 24 KB. The larger the disk operation size, the greater the throughput because fewer disk operations are required to access a given amount of data.

You can increase the total bandwidth of the connection between an xSeries or Blade server and the hosting i5/OS partition by adding target and initiator HBA ports. Note that the number of target HBAs that are connected to a particular xSeries or Blade server must be equal to or greater than the number of initiator HBA ports in the xSeries or Blade server. You can install up to four initiator HBA ports in an xSeries server (one port per HBA adapter) and up to two initiator HBA ports in a Blade server (one or two ports activated on a single HBA adapter). As you add initiator HBA ports to an xSeries server, you can expect the disk I/O throughput to scale fairly linearly, assuming that the xSeries server hardware (processors, memory, PCI bus, and so on) does not become a bottleneck. However, on a Blade server, both initiator HBA ports reside on the same physical adapter. Therefore, you might find that activating the second port on a Blade initiator HBA does not provide twice the throughput.

At the time of writing, there were no performance data or capacity planning guidelines available for Virtual Ethernet LAN. Therefore, in this chapter, we concentrate our discussion of performance and capacity planning on disk storage (SCSI) traffic. It should be noted that in a typical System i-based iSCSI network, most of the I/O traffic results from disk operations, not Virtual Ethernet LAN communication.

Consult the following Web site for the latest performance information:

<http://www.ibm.com/systems/i/bladecenter/iscsi/>

Test environment

There are two approaches you can take to setting up your test environment:

► Non-System i-based test environment

Obviously, it is preferable to complete your capacity planning before you order your new System i iSCSI hardware. This way you can hopefully order what you need in one order without having to reorder additional equipment later on. In this case, you need to set up a test environment that is comparable to a System i iSCSI network.

The Windows Performance Monitor measurements are only useful if they are measured on xSeries or Blade servers that are of a similar hardware specification to the proposed hosted servers. For example, if you are measuring the disk I/O throughput of an old stand-alone server with a single disk drive, then this will not provide useful results. This is because a powerful new xSeries or Blade server connected to System i using iSCSI will likely be able to drive much more disk (and Virtual Ethernet LAN) I/O. You therefore need to allow for differences in the performance of the test servers and the proposed iSCSI attached servers. This can be difficult and result in erroneous results.

► **System i-based test environment**

In reality, the best way to model your proposed iSCSI environment is to use a System i, and xSeries or Blade servers that are of a very similar specification to the proposed hardware. Otherwise, you are likely to obtain misleading results.

There are a number of ways you might be able to access a System i test environment including:

– **Obtain a loan System i from IBM**

Contact your local IBM representative to see if such a system is available.

– **Use your existing System i**

If you have an existing System i machine and you plan to use this machine to connect your *iSCSI network*, then another option is to purchase enough extra hardware to create a prototype environment. In this way, you could perform the capacity planning tasks and then buy additional hardware to build the production iSCSI environment.

– **Purchase a System i from IBM**

If you already plan to buy a new System i, or upgrade your current system, you could buy a System i with enough hardware to perform the capacity planning tasks as a first step, then buy additional hardware after you have calculated exactly what you need.

Either way you need to use xSeries or Blade servers that have very similar configurations to the ones that you will ultimately connect to your System i using iSCSI. Also, you must perform your capacity planning on a system that has a disk configuration that will provide meaningful results. That is, there must be sufficient disk arms to provide adequate performance for the test environment. Even so, it is difficult to exactly model a production environment without actually running the full production environment. Therefore, it is important to err on the conservative side and allow some “fat” or extra resources in the proposed configuration.

We recommend the purchase of at least two target and two initiator HBAs (one for Blade) for your test environment in case you have a Windows server that performs heavy I/O.

You need to make sure that the System i iSCSI test environment is not constrained in terms of the following resources:

– **System i CPW and memory**

The test System i must have sufficient CPW and memory available so that it is effectively running unconstrained as far as these resources are concerned. Unconstrained in this context means that performance of the hosted server is not being impaired by a lack of CPW and/or memory resources. You can use the iSeries Navigator performance monitor graphs to check CPW and memory usage in the hosting partition.

– **System i disk storage**

The main objective of the capacity planning testing is to measure the number of Windows disk operations per second being driven by the hosted server. Therefore, we need a disk configuration on the System i that can absorb all the disk I/Os that are being driven by the Windows server. In other words, there need to be enough disk drives in the System i hosting partition so that disk performance is not a constraint on the performance of the Windows server. You can use the iSeries Navigator performance monitor graphs to check disk activity levels in the hosting partition.

– **Target and initiator HBA ports**

In the case of a heavily configured Windows server running an application that generates many disk I/Os, you need to make sure that the iSCSI HBAs are not causing a bottleneck. We recommend that you initially dedicate a target-initiator HBA port pair

to the test server, or if you know that the server will generate heavy I/O, two or more target-initiator HBA port pairs to the test server. Then use Windows Performance Monitor to determine the number of disk I/Os and the average disk operation size of the Windows workload. You can use this information to calculate the disk I/O throughput for the hosted server in MB/second. Depending on the average disk operation size, you can determine how close to saturation the target and initiator HBAs are, given that an iSCSI HBA has a maximum throughput of approximately 90 MB/second for a disk operation size of 8 KB.

9.11.2 Getting started with capacity planning for iSCSI

In order to accurately size the System i iSCSI environment, you need to first obtain the following performance data for your Windows applications for each planned production server:

- ▶ Disk and Virtual Ethernet LAN throughput in MB/sec
- ▶ Average disk operation size in KB
- ▶ Disk operations per second

Once you have this information, you can calculate the following System i hardware requirements:

- ▶ Number of target HBAs
- ▶ Processor CPW requirements
- ▶ Memory requirements
- ▶ Disk storage requirements

Therefore, there are essentially four steps involved in capacity planning for iSCSI:

1. Obtain Windows performance data

The first step is to measure the number of disk operations per second, the average size of the disk operations in KB, the disk data throughput in MB/sec, and Virtual Ethernet LAN throughput in MB/sec that are generated by each hosted server. Windows Performance Monitor is the best tool to calculate this.

Refer to 9.11.3, “Obtain Windows performance data” on page 392 for a description of how to use the Windows Performance Monitor tool.

2. Calculate the number of target HBAs required

You can use the disk performance data measured for each Windows server to calculate the number of target HBAs required for each Windows server, and thus the total number of target HBAs.

By implication, the number of initiator HBA ports you need for a particular hosted server is equal to the number of target HBAs you calculated for that server.

Refer to 9.11.4, “Calculate the number of target HBAs required” on page 395 for a description of how to perform this calculation.

3. Calculate CPW requirements

Once you know the total number of disk operations per second, you can calculate the CPW requirements to support the iSCSI network.

Refer to 9.11.5, “Calculate CPW requirements” on page 396 for a description of how to perform this calculation.

4. Calculate memory requirements

To calculate the memory requirements of the System i iSCSI environment, you need to know the total number of target HBAs required and the number of active NWSDs. (There is one NWSD per Windows server instance.)

Refer to 9.11.6, “Calculate memory requirements” on page 397 for a description of how to perform this calculation.

5. Calculate disk requirements

The amount of System i disk space needed to support the total disk requirements for the hosted servers is equal to the total requirements of all the individual Windows servers added together. There is no adjustment factor required. However, to take full advantage of the System i SAN architecture, you might wish to change the number and size of your Windows disk drives to take full advantage of the System i's storage virtualization capability.

Refer to 9.11.7, “Calculate disk requirements” on page 398 for more information.

9.11.3 Obtain Windows performance data

This data enables you to calculate the number of target HBAs required to support the disk I/O workload of the hosted Windows servers, and the CPW requirements for the hosted servers.

Note that the number of I/Os generated by the point-to-point *Virtual Ethernet LAN* is negligible. Therefore, unless you have created additional Virtual Ethernet LANs to communicate from the hosted server to other servers or partitions, you do not need to worry about LAN I/Os. However, if you use *Virtual Ethernet LAN* for heavy data transfer, you do need to allow for this when calculating the total number of I/Os.

Using the Windows Performance Monitor

You use the Windows Performance Monitor (WPM) to model disk and Virtual Ethernet LAN I/O performance for a Windows server. WPM enables you to collect the following I/O statistics directly from a Windows server:

- ▶ The disk data throughput in MB/sec
- ▶ Virtual Ethernet LAN data throughput in MB/sec
- ▶ The average disk operation size in KB
- ▶ The number of disk operations/second

Once you have this information, you can calculate the following System i hardware requirements:

- ▶ Number of target HBAs
- ▶ Processor CPW
- ▶ Memory
- ▶ Disk storage

Starting WPM

Make sure you run WPM for at least fifteen minutes up to one hour at the busiest time of the day so that you capture useful information.

To start WPM and save the data to a log file, follow these steps:

1. Click **Start** → **Administrative Tools** → **Performance**.
2. Expand **Performance Logs and Alerts**.
3. Right-click **Counter Logs** and select **New Log Settings**.
4. Enter a log file name in the New Log Settings dialog and click **OK**.

5. Select **Add Counters**.
 6. On the Add Counters dialog, the Select counters from computer radio button should be selected, and the name of the Windows server should appear by default in the associated field.
 7. Select the **PhysicalDisk** Performance object.
 8. Select the **Select counters from a list** radio button, and select the following counters (click **Add** after selecting each one):
 - **Disk Bytes/sec**
This counter gives you the throughput in bytes per second.
 - **Avg. Disk Bytes/Transfer**
This counter gives you the average disk operation size in bytes.
 - **Disk transfers/sec**
This counter gives you the number of disk operations per second.
 9. Select the **Select instances from a list** radio button, and ensure that **_Total** is highlighted. This selects the iSCSI adapter for logging.
 10. (Optional) Only perform the following optional steps if:
 - You are modelling the Windows I/O requirements on a real System i iSCSI network and not some other physical topology.
 - There is significant Virtual Ethernet LAN activity. In other words, the Windows server you are modelling has one or more Virtual Ethernet LAN connections to the System i hosting partition across the iSCSI network, and there is moderate to heavy traffic. Note that the point-to-point Virtual Ethernet LAN does not generate significant LAN traffic.
- On the Add Counters dialog, select the **Network Interface** Performance object.
11. (Optional) Select the **Select counters from a list** radio button, and select the **Bytes Total/sec** counter (click **Add** after selecting it).
This counter gives you the total throughput of the iSCSI initiator HBA ports in bytes per second. This is the combined total of disk I/Os and Virtual Ethernet LAN I/Os. Note that WPM cannot distinguish between the two types of I/Os.
 12. (Optional) Select the **Select instances from a list** radio button, and select the **[\$\$ iSCSI adapters as seen by Windows]** instance (click **Add** after selecting it).
This selects the iSCSI adapter for logging.
 13. (Optional) Repeat the previous step for any additional iSCSI initiator HBA ports in the xSeries or Blade server.
 14. Click **Close**.
 15. In the Counters window, check that you have selected the correct counters.
 16. Set **Sample data every:** to 15 seconds (the default).
 17. Select the **Log Files** tab.
 18. Click the Log file type drop-down list and select **Text File (Comma delimited)**. This will enable you to manipulate the results in a Microsoft Excel® spreadsheet.
 19. Click **Configure** and select a location for the log file.
 20. We recommend that you set a limit on the size of the log file in case you forget to turn off logging. 1 MB should be sufficient.
 21. Click **OK**.
 22. Select the **Schedule** tab.

23. Set the **Start log** and **Stop log** dialogs according to whether you want to run WPM manually, or schedule the start and stop times.

24. Click **OK**.

25. If you chose to manually start logging, double-click **Counter logs** in the left pane of the Performance window. In the right pane, right-click the counter log you have just configured and select **Start**. Logging the metrics you have selected will now start, and continue until either you stop it manually or it is stopped by the scheduler.

The counter is now started.

Ending WPM

To end the counter, follow these steps:

1. Click **Start** → **Administrative Tools** → **Performance**.
2. Select **Counter Logs**.
3. In the right pane, right-click the log file you want to stop and select **Stop**.

The counter is now ended.

Interpreting the WPM results

After you have ended logging, you can manipulate the results as follows:

1. Copy the log file to a computer with Microsoft Excel installed.
2. We recommend that you graph each metric (Microsoft Excel spreadsheet column) to get a feel for how the metric you are measuring behaves over time. Use your observations of the graphed results to help you with the following steps.
3. To calculate total throughput, you use the Disk Bytes/sec metric or the Bytes Total/sec metric, depending on whether or not you are including Virtual Ethernet LAN traffic for the server you are modelling. If the Virtual Ethernet LAN traffic is significant, you should have specified the Bytes Total/sec through the iSCSI adapter ports for collection by WPM. In this case, use the Bytes Total/sec instead of the Disk Bytes/sec to calculate the total throughput. Either way:
 - a. Choose a period of time (at least 10 minutes), where the metric is at its highest. You need to allow for periods when disk activity is at its maximum if the production iSCSI network is going to provide consistently good performance.
 - b. Calculate the Disk Bytes/sec and Bytes Total/sec values over the time period you have chosen. The Virtual Ethernet LAN traffic is equal to the Bytes Total/sec minus the Disk Bytes/sec.
 - c. Record the values and convert to MB/sec. These are the average values for throughput over the period you selected.

You will use these values to calculate the iSCSI hardware requirements.
4. To calculate the number of disk operations per second, you use the Disk Transfers/sec metric as follows:
 - a. Choose a period of time (at least 10 minutes), where the metric is at its highest. You need to allow for periods when disk activity is at its highest if the production iSCSI network is going to provide consistently good performance.
 - b. Calculate the average Disk Transfers/sec values over the time period you have chosen.
 - c. Record the average value per second.

You will use this value to calculate the iSCSI hardware requirements.

5. To calculate the average disk operation size, you use the Avg. Disk Bytes/Transfer metric. In this case, simply average the metric over the total time you ran the counter. Record the average value and convert to KB.

You will use this value to calculate the iSCSI hardware requirements.

You now have values for the following metrics:

► **Disk and Virtual Ethernet LAN throughput in MB/sec**

This is equivalent to the average Disk Bytes/sec (or the average Bytes Total/sec) converted to MB/sec.

► **Disk operations per second**

This is equivalent to the average Disk Transfers/sec.

► **Disk operation size in KB**

This is equivalent to the Avg. Disk Bytes/Transfer converted to KB.

9.11.4 Calculate the number of target HBAs required

As shown in Figure 9-33 on page 388, the throughput of a target *HBA* depends on the size of the disk operations. The larger the disk operation size, the greater the throughput. Table 9-3 shows the maximum number of disk operations per second and the maximum throughput for each disk operation size shown in Figure 9-33 on page 388.

Table 9-3 Disk I/O maximums for a target HBA

Average disk operation size (KB)	Maximum number of disk operations/second	Maximum throughput for each target HBA (MB/sec)
2	3,000	6
4	13,750	55
8	11,250	90
16	6,125	98
24	4,830	116

To calculate the number of target HBAs required for a single hosted server, follow these steps:

1. Compare the average disk operation size, the number of disk operations per second, and the total throughput for a Windows server that you determined in 9.11.3, "Obtain Windows performance data" on page 392 with table Table 9-3.
2. Using the average disk operation size as your starting point, compare the number of disk operations per second and the maximum throughput you obtained using WPM with the values listed in Table 9-3. Interpret the results as follows:
 - a. If the values you obtained using WPM are significantly less than the corresponding values in Table 9-3 then you can be confident that a single target-initiator adapter pair will handle the I/O requirements for the test server. This will probably be the result for most of your test servers.
 - b. If the values you obtained using WPM are approximately equal to, or even greater than, the corresponding values in Table 9-3, then it is likely that the target HBA is saturated and is limiting the I/O throughput of the test server. In this case, you should add another target-initiator HBA port pair and retest. This would be an unusual result and

you would only see this on a server that is generating heavy disk and/or Virtual Ethernet LAN I/O.

- c. Calculate the number of target HBAs for the test server by dividing the throughput you calculated using WPM by the corresponding value in Table 9-3 on page 395. This will likely be a fraction of an adapter, usually less than one.
3. After you have calculated the target HBA requirements for all your test servers, simply add them together to obtain the total requirements for your System i partition. We recommend that you add in an additional target HBA for every three that you calculate that you need. This will allow for possible inaccuracies in your testing and for the fact that the I/O workload cannot be spread completely evenly between the target HBAs.

It should be noted that when we talk about “the number of target HBAs for each Windows server,” in most cases this number will be less than one, assume 0.3. (It has been found in testing that on average, one target HBA will support three or four hosted servers.) However, in cases where a Windows server is driving a large number of disk I/Os, and/or Virtual Ethernet LAN I/Os, it is possible that a single hosted server might require more than one target HBA (and therefore more than one initiator HBA port) to provide adequate I/O bandwidth. Regardless of how many target HBAs are required for each hosted server, the numbers can be added together to give the total HBA requirements. This assumes that storage and LAN I/Os are spread evenly over each of the target HBAs. Note that there is no automatic workload balancing at either the hosting partition or hosted server end. Therefore, when you assign your storage spaces and Virtual Ethernet LANs to the target HBAs, you need to assign them in such a way as to try and evenly spread the workload across the available HBAs. Otherwise, some HBAs might be overloaded, while others can have spare capacity. Therefore, it is always wise to overestimate the number of target HBAs required.

Initiator HBAs

Once you have calculated the number of target HBAs for a Windows server, you must provide an equivalent number of initiator HBAs for the hosted server. Otherwise, the bandwidth will not be balanced across the iSCSI network. For example, there is no point in providing for two target HBAs for a hosted server but installing only one initiator HBA port in the server.

9.11.5 Calculate CPW requirements

While the disk I/O activity driven by iSCSI is not strictly a “CPW” type load, the CPW estimate is still a useful metric to estimate the amount of System i CPU required for a hosted server or servers.

To estimate the required System i CPW, you need to know the number of disk operations per second that the Windows servers will be driving. Use the Windows Performance Monitor to measure the number of disk operations per second as described in 9.11.3, “Obtain Windows performance data” on page 392. You need to obtain the number of Windows disk operations for the busiest time of the day for each Windows server, and then add up the total number for all servers.

The suggested general rule for calculating CPW requirements for an iSCSI network is:

- 19 CPW for every 100 Windows disk operations/second

For example, if you calculate the hosted Windows servers will be generating 800 disk operations/second, you can estimate the CPW usage as:

$$800/100 \times 19 = 152 \text{ CPW}$$

9.11.6 Calculate memory requirements

The System i memory requirements for iSCSI servers vary, and are based on many configuration choices including the number of LUNs (storage spaces), target HBAs, NWSDs, and so on.

Figure 9-34 shows that System i memory is consumed by active target HBAs and NWSDs from the Machine Pool, Base Pool, and QFPHIS Private Pool. The figures provided were averaged across a large iSCSI configuration. While this might not be the same as your planned configuration, there is unlikely to be a significant difference in memory requirements compared with the modelled configuration.

Memory pool type	For each active target HBA	For each active NWSD
Machine pool	21 MB	1 MB
Base pool	1 MB	0.5 MB
QFPHIS private pool	0.5 MB	1 MB
Total	22.5 MB	2.5 MB

Figure 9-34 System i memory requirements for iSCSI

As shown in Figure 9-35 on page 402, the suggested rules for calculating minimum memory requirements for an iSCSI network are:

- ▶ 22.5 MB for every active target HBA
- ▶ 2.5 MB for every active NWSD (corresponds to a hosted Windows server)

Important: Note that IBM Director Server is required in each System i hosting partition. IBM Director Server requires a minimum of 500 MB in the base pool, which needs to be added to the above numbers.

For example, if there are two target HBAs and four active NWSDs in your iSCSI network, you can estimate the CPW requirements as:

- ▶ $(22.5 \times 2) + (2.5 \times 4) + 500 = 550$ MB

QFPHIS private memory pool

The QFPHIS private memory pool referenced in Figure 9-34 is designed to provide caching space for iSCSI disk I/O operations. Its minimum size is 4 096 KB.

The reason why the QFPHIS memory pool is required is that the performance of applications sharing the same memory pool with iSCSI disk operations might be adversely impacted if the iSCSI-connected Windows servers perform levels of disk I/O which cause flushing of the memory pool. Therefore, it is possible for other applications to begin page faulting because their cached data in memory has been flushed out to disk by the iSCSI I/O operations.

By default, the iSCSI disk I/O operations occur in the *BASE memory pool. In order to segregate iSCSI disk activity from other applications, V5R4 PTF SI23027 has been created to enable iSCSI disk I/O operations to run in a new private memory pool, QFPHIS. This pool is enabled by creating a subsystem description named QGPL/QFPHIS and allocating at least 4 096 KB to it. The amount of memory you need to allocate depends on a number of factors, including the number of active iSCSI-connected Windows servers and the sustained disk activity for all hosted servers, as shown in Figure 9-34.

To create the QFPHIS private memory pool for iSCSI-connected Windows servers, follow these steps:

1. Run the following CL command from a 5250 or green screen command line:

```
CRTSBSD SBSD(QGPL/QFPHIS) POOLS((1 10000 1))
```

This example creates a subsystem with a memory pool of 10,000 KB (~10 MB) in size. 10,000 KB is a reasonable minimum value, but 4,096 KB is the absolute minimum size supported.

2. Vary on the Network Server Description (NWSD) of any iSCSI-connected Windows server to actually activate the memory pool.

During the NWSD vary on processing, the QFPHIS subsystem is automatically started if necessary, and the private memory pool activated. iSCSI NWSDs that are varied on will then utilize the private memory pool for disk I/O operations. The private memory pool is used by the active servers as long as the subsystem remains active. If the QFPHIS subsystem is ended prematurely (while an iSCSI NWSD is active), the servers continue to function properly, but subsequent disk I/O operations revert to the *BASE memory pool. Active iSCSI servers that are varied on and using the QFPHIS memory pool at the time the subsystem is ended can adversely impact other applications either when the memory pool reverts to *BASE or in the event of the memory pool identifier being reassigned to another subsystem. To prevent unexpected performance impacts, do not end the QFPHIS subsystem while iSCSI servers are active.

9.11.7 Calculate disk requirements

The level of disk I/O achieved on the IXS and IXA varies, and it depends on many variables. However, given an adequate storage subsystem, the upper cap on I/O for a single server is limited by the IOP component of the IXS/IXA hardware. Except in extreme test loads, it is unlikely the IOP will saturate due to disk activity. When multiple IXS/IXA servers are configured in the same System i partition, the partition software imposes a cap on the aggregate total I/O from all the servers. It is not a strict limitation, but a typical capacity level is approximately 6 000 to 10 000 disk operations/sec.

The iSCSI implementation uses a more scalable storage I/O access architecture than the IXS and IXA solutions. As a result, a single hosted server can scale to greater disk I/O capacity by using multiple target and initiator HBAs to enable multiple data paths between the server and its hosting partition. In addition, there is no inherent partition cap on the iSCSI disk I/O. The entire performance capacity of installed disks and disk controllers is available to iSCSI attached servers.

iSCSI attached servers use non-reserved System i virtual storage in order to perform disk input or output. Thus, disk operations use System i memory as an intermediate read cache. Write operations are flushed to disk immediately, but the disk data remains in memory and can be read on subsequent operations to the same sectors.

In V5R4, the CHGNWSSTG command and iSeries Navigator now support the enlargement of a Windows virtual disk. After the storage space has been expanded using the iSeries Navigator virtual disk New Based On option, or WRKNWSSTG option 3, the Windows partition also needs to be expanded. The Windows Server 2003 DISKPART command can be used to perform the partition expansion; however, DISKPART only expands the file system on Windows basic disks. If a Windows disk has been converted to dynamic, the DISKPART command creates a new partition and configures a spanned volume across the partitions. With iSCSI, the second partition would experience degraded disk performance.

Recommendations

Here are recommendations to keep in mind when designing your virtual disk configuration for use with System i iSCSI.

► Virtual disk creation rules

With iSCSI, there are a few Windows side disk configuration rules you must take into account to enable efficient disk operations, and to minimize memory fault activity associated with iSCSI disk operations. Windows disks must be configured as follows:

- One Windows partition per virtual drive
- Windows file system formatted with cluster sizes of 4 KB or 4 KB multiples
- 2 GB or larger storage spaces (for which Windows creates a default NTFS cluster size of 4 KB)

These rules allow i5/OS to efficiently manage the storage space memory and mitigate disk operation faulting activity, which improves overall iSCSI disk I/O performance. These rules could also slightly improve IXS and IXA attached servers' disk performance, but to a much smaller degree.

► Sizing virtual disks for performance

Windows virtual disks are created out of System i single level storage, and therefore they are highly virtualized. Each virtual disk is physically spread over all the physical disk drives in the i5/OS ASP. Because of this architecture, the size of a virtual disk is independent of performance. In other words, there is no performance benefit in having a larger number of smaller virtual disks. Therefore, to keep Windows disk administration simple we recommend that you create the minimum number of virtual disks that you need to structure your information optimally.

Note that if you want to confine your virtual disks to a specific i5/OS ASP, you can specify the ASP when you create a virtual disk.

► Assigning a virtual disk to an HBA

To make a virtual disk visible to Windows, you must link it to a target HBA. At the time of writing, you could only link a storage space to one target HBA. IBM intends to provide Multipath I/O capability in a later version of the System i iSCSI code. Because of this limitation, for a hosted server to make use of multiple HBAs (target or initiator), you must have at least one storage space/virtual disk that can be assigned to each target or initiator HBA. For example, assume you determined that two target and initiator HBA ports were required to satisfy the disk I/O bandwidth requirements of a particular hosted server. In this case, you would need to create at least two storage spaces/virtual disks, split the data between them, and link each one to a separate HBA port in order to make use of the two ports.

On the hosted server side, virtual disks are automatically assigned to an initiator HBA port. However, it is possible to exert a degree of control over the assignment of virtual disks to initiator HBA ports by manually disabling certain ports for the attachment of virtual disks.

9.11.8 Scalability limits

Whether you are scaling your iSCSI network by increasing the bandwidth of a single hosted server connection or by increasing the number of hosted servers, there are limits that you need to be aware of.

These limits include:

- ▶ The number of target iSCSI HBAs that can be installed in each model of the System i

The limits are documented on the following Web site:

<https://www.ibm.com/systems/i/bladecenter/iscsi/#support>

- ▶ The maximum number of hosted servers that can be supported by a single target HBA

This number is eight.

- ▶ The number of hosted xSeries and Blade servers that can be connected to each model of the System i

You can intermix xSeries and Blade servers on the System i iSCSI network in any combination. The maximum number of hosted servers that can be connected to a particular model of the System i depends on the previous two points plus the number of hosted servers that can be *practically* supported by each target HBA.

Although up to eight hosted servers can be supported by each target HBA, each server shares the target HBA's 1 Gb bandwidth. For servers that are doing very little disk or Virtual Ethernet LAN I/O, it might be possible to support eight on a single target HBA. At the other extreme, 1 Gb bandwidth might not be enough to support even one server that is doing heavy disk I/O (such as a Microsoft SQL server). Therefore, the practical limit usually lies somewhere between one and eight for a typical Windows server. You need to perform some capacity planning for your Windows servers to determine how many you can practically support on each target HBA.

We recommend a maximum of three to four hosted servers per target HBA, depending on workload.

- ▶ The maximum number of storage spaces that can be linked to an NWSD

This number is 64.

- ▶ The maximum amount of virtual disk storage that can be accessed by a hosted Windows server

This amount is 60 TB.

- ▶ The maximum number of Virtual Ethernet LANs that can be connected to an NWSD

This number is five and includes the Point-to-point Virtual Ethernet LAN.

- ▶ The maximum number of target HBAs that can be used by a hosted server to provide storage space links

This number is four. These target HBAs could be the same as those used to provide Virtual Ethernet LAN connections, or they could be different.

- ▶ The maximum number of target HBAs that can be used by a hosted server to provide Virtual Ethernet LAN connections

This number is four. These target HBAs could be the same as those used to provide storage space links, or they could be different.

- ▶ The maximum number of initiator HBA ports that can be supported on an xSeries server

This number is four. Each initiator HBA that can be installed in an xSeries server has one port only.

- ▶ The maximum number of initiator HBA ports that can be supported on a Blade server

This number is one. Each initiator HBA that can be installed in a Blade server has two ports.

- ▶ The maximum number of storage spaces that can be supported by a target HBA

This number is $64 \times 8 = 512$ where 64 is the maximum number of storage spaces per hosted server, and eight is the maximum number of hosted servers per target HBA.

- ▶ The maximum number of Virtual Ethernet LANs that can be supported by a target HBA
This number is eight.

- ▶ The maximum combined number of storage spaces and Virtual Ethernet LANs that can be supported by a target HBA

This number is $512 + 8$. It is the cumulative total of the number of storage spaces and Virtual Ethernet LANs that can be supported by a target HBA.

- ▶ The maximum number of virtual disks that can be supported by an initiator HBA port

This number is 64 and is the same as the maximum number of virtual disks that can be supported by a hosted server.

- ▶ The maximum number of Virtual Ethernet LANs that can be supported by an initiator HBA port

This number is five and is the same as the maximum number of Virtual Ethernet LANs that can be supported by a hosted server.

- ▶ The maximum combined number of virtual disks and Virtual Ethernet LANs that can be supported by an initiator HBA port

This number is $64 + 5$. It is the cumulative total of the number of virtual disks and Virtual Ethernet LANs that can be supported by a hosted server.

9.12 IP addressing in a System i iSCSI network

In this section, we discuss how IP addressing works in a System i iSCSI network. We cover the following topics:

- ▶ **iSCSI network addressing**

We explain how IP addresses are used in an iSCSI network to establish communication between the target and initiator HBA ports.

Refer to 9.12.1, “iSCSI network addressing” on page 401.

- ▶ **Choosing an IP addressing schema**

We put forward a sample IP addressing schema that you can use in your iSCSI network.

Refer to 9.12.2, “Choosing an IP addressing schema” on page 404.

9.12.1 iSCSI network addressing

A System i iSCSI network is really a TCP/IP network, which runs over a 1 Gb switched Ethernet LAN topology. Therefore, each node on the iSCSI network requires an IP address to communicate with other nodes on the network. In the System i iSCSI network, the nodes are target HBAs and initiator HBA ports. Therefore, all HBA ports require IP addresses.

Figure 9-35 on page 402 shows a schematic of a basic iSCSI configuration where a single target HBA is communicating with a single initiator HBA port across the iSCSI network.

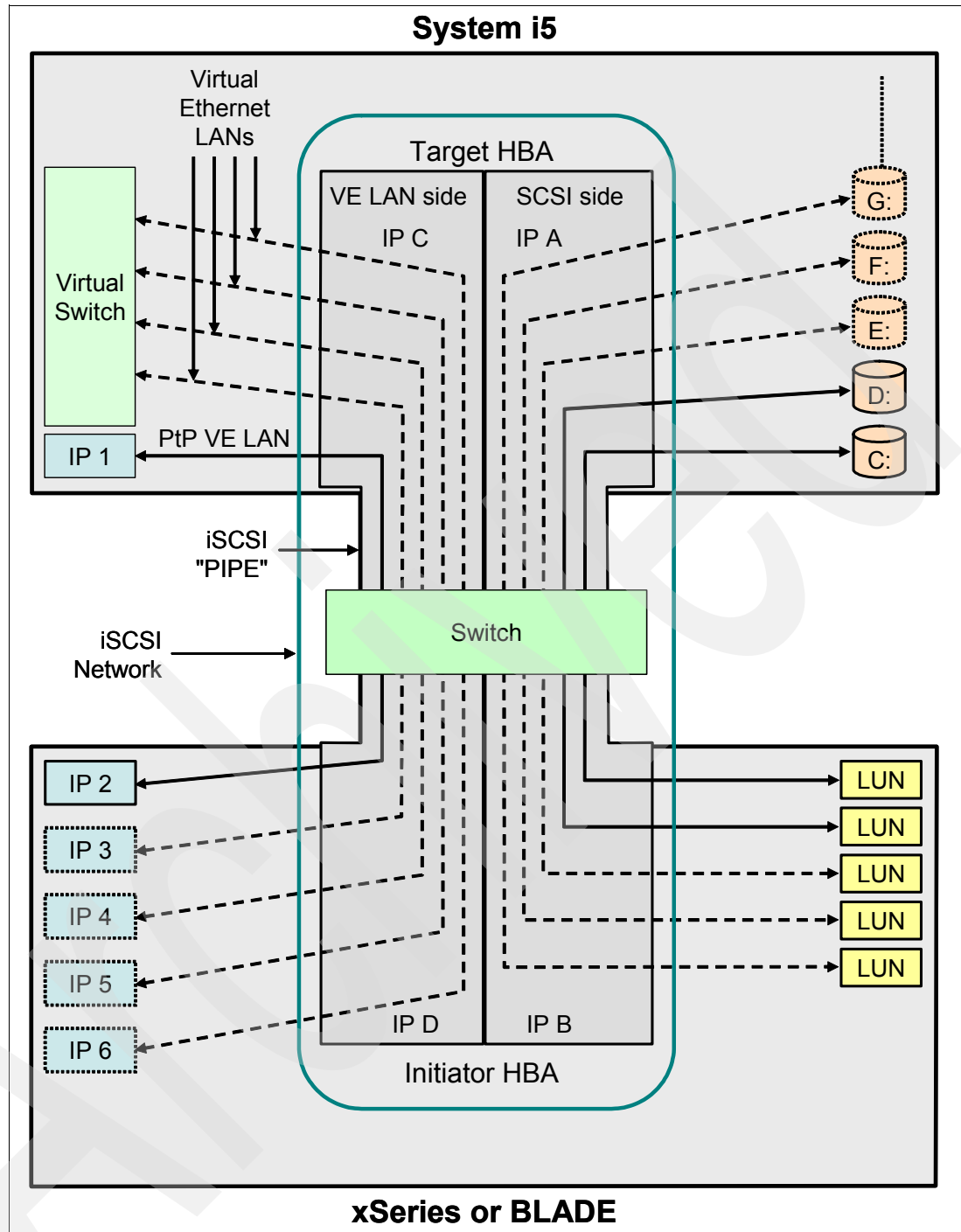


Figure 9-35 iSCSI network IP addressing structure

In Figure 9-35 note the following points:

- ▶ Every iSCSI HBA contains its own TCP/IP stack, implemented in hardware, that is independent of the normal i5/OS TCP/IP stack.
- ▶ The data path between the target and initiator HBA ports is split into two “channels”, one for the transport of SCSI (virtual disk) traffic, the other for the transport of LAN (Virtual Ethernet LAN) traffic.

- ▶ Each channel has an IP address at both ends. In Figure 9-35 on page 402, they are IP A and IP B for the SCSI channel, and IP C and IP D for the LAN channel.
- ▶ On the SCSI side of the data path, the storage spaces in the i5/OS partition exchange data with the hosted Windows server, which sees the storage spaces as LUNs (virtual disks). The C: and D: (boot) drives are created automatically when a hosted server is installed. You can create additional storage spaces (E:, F:, G:.....) and link them for use by Windows running on the hosted server.
- ▶ On the LAN side of the data path, the Virtual Ethernet LAN endpoints enable the hosted Windows server to communicate with other hosted servers, or with partitions on the System i. Each endpoint on a Virtual Ethernet LAN must have an IP address, just like a real Ethernet LAN. There are two types of Virtual Ethernet LAN:

- **Point-to-point Virtual Ethernet LAN**

The point-to-point Virtual Ethernet LAN is created automatically when a hosted server is installed. The IP addresses for the endpoints of this LAN are automatically generated by i5/OS when the server is created, although you can manually specify them if you wish. In Figure 9-35 on page 402, these addresses are IP 1 and IP 2.

- **Virtual Ethernet LAN**

Up to four additional Virtual Ethernet LANs can be created. These non-point-to-point Virtual Ethernet LANs are simply referred to as Virtual Ethernet LANs. The IP addresses for the endpoints of these LANs must be manually configured. Note that each Virtual Ethernet LAN connects to a virtual switch in the i5/OS partition. This is analogous to a real (physical) switched Ethernet LAN where all the network nodes are connected by a switch. The point-to-point Virtual Ethernet LAN, as the name implies, is a direct connection between the hosted server and its hosting partition, and therefore does not connect to the virtual switch.

- ▶ The target HBA IP addresses (IP A and IP C) are specified in the NWSH object, which describes the target HBA. If you want to use different IP addresses, you must change the NWSH configuration and restart the NWSH. Note that the i5/OS TCP/IP stack is unaware of the IP addresses configured for the target HBAs.
- ▶ The initiator HBA port IP addresses (IP B and IP D) are specified in the RMTSYS object, which describes the hosted server. The RMTSYS object differs from the NWSH object in several ways:
 - You can configure an initiator HBA port with 1 or 2 IP addresses; in other words, you can leave one of the IP addresses, SCSI or LAN, blank. The reason you might want to do this relates to path deployment considerations as described in 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
 - Whenever you configure an IP address for an initiator HBA port, you must also configure the corresponding port’s MAC address. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter. An xSeries HBA has a label with two MAC addresses (one for the SCSI side and the other for the LAN side) whereas a Blade HBA has a label with four MAC addresses (one for the SCSI side and the other for the LAN side of each port). The SCSI side MAC address is labelled “iSCSI” and the LAN side is labelled “TOE” (TCP Offload Engine). Note that you need to record the MAC addresses on the Blade HBA, before you install it because the sticker is not visible once the HBA is installed.
 - You configure all of the initiator HBA ports for a hosted server in the same RMTSYS object. When the hosted server is subsequently powered on, i5/OS Integrated Server Support automatically ensures that initiator HBA ports in the hosted server are using the IP addresses specified in the RMTSYS object. If you want to use different IP addresses, you must change the RMTSYS configuration and restart the server.

- SCSI traffic uses the initiator HBA port's hardware TCP/IP stack, but LAN traffic uses the Windows TCP/IP stack. Therefore, the Windows TCP/IP stack is unaware of the SCSI IP address, but it is aware of the LAN IP address. This enables you to redeploy Virtual Ethernet LANs from one initiator HBA port to another through Windows as described in 9.15.5, "Redeploying a Virtual Ethernet LAN to a different initiator HBA port" on page 446. Note that virtual disks cannot be redeployed in this way.

9.12.2 Choosing an IP addressing schema

When you configure target and initiator HBA ports, you need to enter IP addresses in the NWSH and RMTSYS objects. In Figure 9-35 on page 402, we represent these addresses as IP A, IP B, IP C, and IP D.

The more target and initiator HBAs you have, the more IP addresses you need to configure. It is important that you allocate IP addresses in a systematic and consistent way. For this reason, we have included a recommended IP addressing schema as shown in Figure 9-36.

	Target HBA port 1	Target HBA port 2	Target HBA port 3
SCSI address:	128.168.201.1	128.168.201.2	128.168.201.3
Subnet mask:	255.255.0.0	255.255.0.0	255.255.0.0
Gateway address:	128.168.200.1	128.168.200.1	128.168.200.1
LAN address:	128.168.202.1	128.168.202.2	128.168.202.3
Subnet mask:	255.255.0.0	255.255.0.0	255.255.0.0
Gateway address:	128.168.200.1	128.168.200.1	128.168.200.1
	Initiator HBA port 1	Initiator HBA port 2	Initiator HBA port 3
SCSI address:	128.168.203.1	128.168.203.2	128.168.203.3
Subnet mask:	255.255.0.0	255.255.0.0	255.255.0.0
Gateway address:			
LAN address:	128.168.204.1	128.168.204.2	128.168.204.3
Subnet mask:	255.255.0.0	255.255.0.0	255.255.0.0
Gateway address:			

Figure 9-36 Recommended IP addressing schema

In Figure 9-36, note the following points:

- ▶ These are Class B addresses and therefore have a subnet mask of 255.255.0.0. rather than the more familiar Class C subnet mask of 255.255.255.0. A Class B subnet allows you to have all addresses used in the sample schema on the same subnetwork, which simplifies the addressing structure.
- ▶ For each target or initiator HBA port, you enter a SCSI and a LAN IP address. The SCSI side of an HBA is used to provide a conduit for disk I/Os, while the LAN side of an HBA is used to provide a conduit for Virtual Ethernet LAN I/Os. Therefore, you need two addresses for each HBA port.
- ▶ You must include a gateway address when you are configuring a target HBA, although it is not used.
- ▶ You must *not* include a gateway address when you are configuring an initiator HBA port.

We map IP addresses shown in Figure 9-36 on page 404 to Figure 9-35 on page 402 as follows:

- ▶ A target HBA SCSI address is equivalent to IP A.
- ▶ A target HBA LAN address is equivalent to IP C.
- ▶ An initiator HBA port SCSI address is equivalent to IP B.
- ▶ An initiator HBA port LAN address is equivalent to IP D.

To use the schema, you simply enter the IP addressing information shown in Figure 9-36 on page 404 in the NWSH object (in the case of a target HBA), or the RMTSYS object (in the case of an initiator HBA port). You do need to keep track of which addresses you have already used, because the addressing information for each target and initiator HBA port must be unique.

9.13 Creating an iSCSI connections map

There is a tool, called QVNIMAP, which generates a “map” showing data path information for target and initiator HBA ports for a hosted Windows server. QVNIMAP runs under Windows and is provided as part of the System i Integration for Windows Server software (V5R4).

The output of the QVNIMAP tool can be used to determine the current distribution of storage space and Virtual Ethernet LAN connections across the available target and initiator HBAs. You can use this information to help manage the connections, and plan how you can balance the number of I/Os flowing across each HBA. Note that the tool merely provides a map of the existing connections, it does not give you any recommendations as to the best way to distribute the connections to balance data flow across the target and initiator HBAs.

Note that the current version of QVNIMAP does not display tape or optical devices accessed via iSCSI.

We cover the following topics:

- ▶ **Running the QVNIMAP command**

We show the format of the command, the switches that you can specify, and we provide examples of how you can use the command.

Refer to 9.13.1, “Running the QVNIMAP command” on page 405.

- ▶ **Using the QVNIMAP switches**

Here we provide sample output for each of the QVNIMAP switches and describe what the data means.

9.13.1 Running the QVNIMAP command

You run QVNIMAP /? from a command line on a hosted Windows server as shown in Figure 9-37 on page 406.

```

C:\>qvnimap /?
Displays target and initiator port usage for storage and Virtual Ethernet.

qvnimap [ /? | [/stg | /lan | /stgcnn | /lancnn | /inrcfg |
              /inrstgadr | /inrstgiqn | /inrlanadr | /inrlannam |
              /tgtstgadr | /tgtstgiqn | /tgtlanadr] [/csv] ]

/?          Display this help message.
/stg         All tables related to storage.
/lan        All tables related to Virtual Ethernet (LAN).
/stgcnn     Storage Device Connections table.
/lancnn     Virtual Ethernet (LAN) Adapter Connections table.
/inrcfg     Initiator Configuration Details
/inrstgadr  Initiator SCSI Details -- Address table.
/inrstgiqn  Initiator SCSI Details -- IQN table.
/inrlanadr  Initiator LAN Details -- Address table.
/inrlanad  Initiator LAN Details -- Description table.
/inrlannam  Initiator LAN Details -- Connection Name table.
/tgtstgadr  Target SCSI Details -- Address table.
/tgtstgiqn  Target SCSI Details -- IQN table.
/tgtlanadr  Target LAN Details -- Address table.
/csv       Output using comma-separated values.
           o Table titles are omitted.
           o Table column headings are limited to one line.

```

Figure 9-37 Running the QVNIMAP tool from a Windows command line

The output of the QVNIMAP command can either be displayed on the Windows server's console, or redirected to a file. Here are several examples of the output that you can receive from this command using the various switches:

► **Display all information:**

```
C:\>qvnimap
```

This example displays the output of all switches.

► **Display a subset:**

```
C:\>qvnimap /tgtstgadr
```

This example displays IP address information for the SCSI side of each target HBA only.

► **Redirect all information to a text file (.TXT):**

```
C:\>qvnimap > c:\output1.txt
```

This example redirects the output of all switches to the file output1.txt.

► **Redirect all information to a comma-separated values file (.CSV):**

```
C:\>qvnimap /stg /csv > c:\output2.csv
```

This example redirects all iSCSI storage-related information to the file output2.csv in comma-separated value format.

9.13.2 Using the QVNIMAP switches

Here is a description of the output of each of the QVNIMAP switches as shown in Figure 9-37.

/stg

This switch displays information that relates to the SCSI side of the target and initiator HBA ports. All of the tables that contain “stg” in the switch name are displayed when this switch is specified.

/lan

This switch displays information that relates to the LAN side of the target and initiator HBA ports. All of the tables that contain “lan” in the switch name are displayed when this switch is specified.

/stgcnn

The /stgcnn switch shows the storage spaces created for the hosted server, the storage paths that they are linked to, and the physical connections used between the initiator HBA ports and the NWSHs. An example is shown in Figure 9-38.

Storage Device Connections:									
Disk	Drive	LUN	Name	Path	-----Target-----	--Initiator--			
					i5/OS NWSH	Connections			
						P1	P2	P3	P4
0	C:	0	VDASD_ETNWS111	1	HNWSH4		X		
1	D:	1	VDASD_ETNWS112	1	HNWSH4		X		
2		2	VDASD_DISK0001	2	HNWSH5				X
3		3	VDASD_DISK0002	3	HNWSH6	X			
None	N/A	N/A	N/A	4	HNWSH7			X	

Figure 9-38 QVNIAMAP /stgcnn switch

Read Figure 9-38 as follows:

► Disk

This is the Windows disk sequence number as shown in the Windows Disk Management snap-in: **My Computer** → **Manage** → **Disk Management**.

► Drive

This is the Windows drive letter that is assigned to the storage device.

The blanks mean that the storage device has not been assigned a drive letter. There are a number of reasons why this might be, including:

- The storage space has not yet been formatted.
- The storage space has been formatted but the user has not yet assigned a drive letter.
- The storage space has been mounted to a Windows directory.

► LUN

This is the LUN (logical unit number) in Windows. A LUN is a SCSI concept. Each device on a SCSI bus is assigned a unique identifier to distinguish it from other devices that share the same bus. To find the LUN number, start Windows Explorer, right-click one of the disk drive icons and select Properties. The LUN number can be found on the Hardware tab.

Note that the LUN is not necessarily the same as the Disk number.

► Name

This is the name of the storage space. The names for the system and install drives (usually these are the C: and D: drives) are automatically generated when you install the server, and are based on the name you select for the NWSD. Other storage space names are defined by the user when they are created.

► **Target i5/OS Path**

This is a distinguishing number for each storage path that has been created for an NWSH. This number corresponds to the number listed in the Path column under the Storage Paths tab of the hosted server's NWSD Properties.

► **Target i5/OS NWSH**

Each NWSH represents a target HBA. In Figure 9-38 on page 407, note the following points:

- There are two storage spaces linked to NWSH4 via storage path 1. The system and install drives must always be linked to the same storage path.
- No disks have been linked to NWSH7, so there are no storage space or virtual disk details.

You can tie an NWSH back to a physical target HBA as follows:

- Determine the HBA's hardware resource ID specified in the NWSH (LINxx).
- Run the CL command WRKHDWRSC *CMN in a green screen.
- Enter a 7 against the LINxx resource in WRKHDWRSC *CMN.
- The Frame ID and Card position enable you to locate the HBA in a tower.

► **Initiator Connections P1 - P4**

The mini-table shown in Figure 9-38 on page 407 shows the storage data paths that have been established between the hosting partition and this hosted server. In other words, it shows the data paths that have been established between initiator HBA ports (P1 - P4) and the target HBAs represented by NWSH4, 5, 6, and 7. You also see which storage path is used by each storage space.

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-41 on page 410 to determine the SCSI MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

/lancnn

The /lancnn switch shows the Virtual Ethernet LANs created for the hosted server, their IP addresses, and the target and initiator HBA ports that the Virtual Ethernet LANs are connected to. An example is shown in Figure 9-39.

Virtual Ethernet (LAN) Adapter Connections:			--Target--	--Initiator--			
Port	Virtual Adapter IP Address		i5/OS NWSH	P1	P2	P3	P4
VRTETHPTP	192.168.103.2		HNWSH5			X	
VRTETH0	192.168.0.3		HNWSH6				X
VRTETH1	192.168.1.3		HNWSH7	X			
VRTETH2	192.168.2.3		HNWSH4		X		
VRTETH3	192.168.3.3		HNWSH5			X	

Figure 9-39 QVNIMAP /lancnn switch

Read Figure 9-39 as follows:

► **Port**

This is the short name for a Virtual Ethernet LAN. Note that VRTETHPTP is a stand-alone point-to-point Virtual Ethernet LAN between the hosted server and hosting i5/OS partition.

► **Virtual Adapter IP address**

This is the IP address of the Windows end of the Virtual Ethernet LAN connection.

► **Target i5/OS NWSH**

This represents the target HBA to which the Virtual Ethernet LAN is assigned.

You can tie an NWSH back to a physical target HBA as follows:

- Determine the HBA's hardware resource ID specified in the NWSH (LINxx).
- Run the CL command WRKHDWRSC *CMN in a green screen.
- Enter a 7 against the LINxx resource in WRKHDWRSC *CMN.
- The Frame ID and Card position enable you to locate the HBA in a tower.

► **Initiator Connections P1 - P4**

The mini-table shown in Figure 9-39 on page 408 shows the LAN data paths that have been established between the hosting partition and this hosted server. In other words, it shows the data paths that have been established between initiator HBA ports (P1 - P4) and the target HBAs represented by NWSH4, 5, 6, and 7. You also see which data path is used by each Virtual Ethernet LAN.

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-43 on page 412 to determine the LAN MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

/inrcfg

The /inrcfg switch displays configuration information about the initiator HBA ports installed in the hosted server. An example is shown in Figure 9-40.

Initiator Configuration Details:					
Port	Type	MTU	BIOS	Firmware	Boot Mode
P1	QLA4050	1500	1.08	2.0.0.29	Disable
P2	QLA4050C	1500	1.08	2.0.0.29	DHCP
P3	QLA4050C	1500	1.08	2.0.0.29	Disable
P4	QLA4050	1500	1.08	2.0.0.29	Disable

Figure 9-40 QVNI MAP /inrcfg switch

Read Figure 9-40 as follows:

► **Port**

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-41 on page 410 to determine the SCSI MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

► **Type**

This is the type of physical initiator HBA port that is installed in the hosted server. At the time of writing, the supported adapters were:

- QLA4050 is the fiber HBA for xSeries.
- QLA405C is the copper HBA for xSeries.
- QMC4052 is the dual port HBA for Blade. Note that QMC4052 is displayed for both ports on the one adapter. Also, because the BladeCenter switch module (or external switch in the case of a passthru adapter) provides the physical network connectivity (copper or fiber), there is only one type of adapter for Blade.

► **MTU**

The Maximum Transmission Unit is set directly on the initiator HBA port using the Fast!UTIL utility. Although this value can be set to 1 500 or 9 000, we recommend 1 500.

► **BIOS**

This is the current level of the BIOS on the initiator HBA. For information about the required level, refer to the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/servermodels/>

► **Firmware**

This is the current firmware level of the initiator HBA. For information about the required level, refer to the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/servermodels/>

► **Boot Mode**

The boot mode is set on the initiator HBA port using the Fast!UTIL utility.

/inrstgadr

The /inrstgadr switch displays MAC and IP address information for the SCSI side of each initiator HBA port. An example is shown in Figure 9-41.

Initiator SCSI Details -- Address table:			
Port	i5/OS RMTIFC	MAC Address	IP Address
P1	1	00-0D-60-BC-03-9D	192.168.99.115
P2	2	00-C0-DD-03-F3-72	192.168.99.111
P3	3	00-0D-60-BC-03-6D	192.168.99.113
P4	4	00-0D-60-BC-03-E7	192.168.99.117

Figure 9-41 QVNI MAP /inrstgadr switch

Read Figure 9-41 as follows:

► **Port**

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-41 to determine the SCSI MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

► **i5/OS RMTIFC**

Numbers 1 - 4 correspond to the numbers in the Interfaces column under the Network Interfaces tab of the RMTSYS Properties. They often correspond to P1 - P4, but not necessarily.

► **MAC Address**

The MAC addresses shown here belong to the SCSI side of the initiator HBA ports in the hosted server. The SCSI and LAN sides of the iSCSI HBA ports each have their own unique MAC addresses.

An xSeries HBA has a label with two MAC addresses (one for the SCSI side and the other for the LAN side) whereas a Blade HBA has a label with four MAC addresses (one for the SCSI side and the other for the LAN side of each port). The SCSI side MAC is labelled "iSCSI" and the LAN side is labelled "TOE" (TCP Offload Engine). Note that you need to record the MAC addresses on the Blade HBA before you install it, because the sticker is not visible once the HBA is installed.

► **IP Address**

The IP addresses shown here belong to the SCSI side of the initiator HBA ports in the hosted server. Note that the SCSI and LAN sides of the iSCSI HBA ports each have their own unique IP addresses. For more information about IP addressing in a System i iSCSI network, refer to 9.12, “IP addressing in a System i iSCSI network” on page 401.

/inrstgiqn

The /inrstgiqn switch displays the IQNs for the SCSI side of each initiator HBA port. An example is shown in Figure 9-42.

Initiator SCSI Details -- IQN table:		
Port	i5/OS RMTIFC	iSCSI Qualified Name (IQN)
P1	1	iqn.1924-02.com.ibm:23a0808.i2
P2	2	iqn.1924-02.com.ibm:23a0808.i0
P3	3	iqn.1924-02.com.ibm:23a0808.i1
P4	4	iqn.1924-02.com.ibm:23a0808.i3

Figure 9-42 QVNiMAP /inrstgiqn switch

Read Figure 9-42 as follows:

► Port

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-41 on page 410 to determine the SCSI MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

► i5/OS RMTIFC

Numbers 1 - 4 correspond to the numbers in the Interfaces column under the Network Interfaces tab of the RMTSYS Properties. They often correspond to P1 - P4, but not necessarily.

► iSCSI Qualified Name (IQN)

Each *initiator HBA* SCSI port not only has a unique MAC and IP address, but it also has a unique IQN. In the wider storage context, an IQN is an industry standard term that uniquely represents a storage area network (SAN) interface.

Initiator IQNs can either be user-defined or automatically generated by i5/OS. However, unless you have a specific reason, we recommend that you let them be automatically generated.

For more information about IQNs, refer to the following Web site:

<http://www.ietf.org/rfc/rfc3720.txt>

The initiator HBA SCSI port IQN is comprised of several parts. For example, the IQN in Figure 9-42 for P1 is:

iqn.1924-02.com.ibm:23a0808.i2

The components of this IQN and their meanings are as follows:

– **iqn**

The iqn prefix designates this string as an industry standard IQN.

– **1924-02**

The date code 1924-02 is based on the year and month when IBM starting using IBM as the company name.

– **.com.ibm**

.com.ibm is equivalent to a domain suffix for a storage area network (SAN).

– **23a0808**

This string is the xSeries or Blade server serial number (with any letters in lower case).

– **i2**

These characters represent “i” for *initiator* followed by a number that distinguishes this IQN from all others on this xSeries or Blade server.

/inrlanadr

The /inrlanadr switch displays MAC and IP address information for the LAN side of each initiator HBA port as well as MAC and IP address information for each Virtual Ethernet LAN network connection in Windows. An example is shown in Figure 9-43.

Initiator LAN Details -- Address table:			
Port	i5/OS RMTIFC	MAC Address	IP Address
P1	1	00-0D-60-BC-03-9C	192.168.99.116
P2	2	00-C0-DD-03-F3-71	192.168.99.112
P3	3	00-0D-60-BC-03-6C	192.168.99.114
P4	4	00-0D-60-BC-03-E6	192.168.99.118
VRTETHPTP		FA-F6-F2-31-08-08	192.168.103.2
VRTETH0		FA-E6-F2-31-08-08	192.168.0.3
VRTETH1		FA-D6-F2-31-08-08	192.168.1.3
VRTETH2		FA-C6-F2-31-08-08	192.168.2.3
VRTETH3		FA-B6-F2-31-08-08	192.168.3.3

Figure 9-43 QVNIMAP /inrlanadr switch

Read Figure 9-43 as follows:

► **Port**

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-43 to determine the LAN MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

VRTETHxxx is the short name for a Virtual Ethernet LAN. Note that VRTETHPTP is a stand-alone point-to-point Virtual Ethernet LAN between the hosted server and hosting i5/OS partition.

► **i5/OS RMTIFC**

Numbers 1 - 4 correspond to the numbers in the Interfaces column under the Network Interfaces tab of the RMTSYS Properties. They often correspond to P1 - P4, but not necessarily.

► **MAC Address**

The first four MAC addresses shown here belong to the LAN side of the initiator HBA ports in the hosted server. The SCSI and LAN sides of the iSCSI HBA ports each have their own unique MAC addresses.

An xSeries HBA has a label with two MAC addresses (one for the SCSI side and the other for the LAN side) whereas a Blade HBA has a label with four MAC addresses (one for the SCSI side and the other for the LAN side of each port). The SCSI side MAC is labelled “iSCSI” and the LAN side is labelled “TOE” (TCP Offload Engine). Note that you need to record the MAC addresses on the Blade HBA before you install it because the sticker is not visible once the HBA is installed.

The last five MAC addresses shown here are MAC addresses for each Virtual Ethernet network connection in Windows.

► **IP Address**

The first four IP addresses shown here belong to the LAN side of the initiator HBA ports in the hosted server. Note that the SCSI and LAN sides of the iSCSI HBA ports each have their own unique IP addresses. For more information about IP addressing in a System i iSCSI network, refer to 9.12, “IP addressing in a System i iSCSI network” on page 401.

The last five IP addresses shown here are IP addresses for each Virtual Ethernet network connection in Windows.

/inrland

The /inrland switch displays descriptive information for each Virtual Ethernet LAN network connection in Windows. An example is shown in Figure 9-44.

Initiator LAN Details -- Description table:		
Port	i5/OS RMTIFC	Description
P1	1	QLogic 1Gb PCI Ethernet Adapter
P2	2	QLogic 1Gb PCI Ethernet Adapter #3
P3	3	QLogic 1Gb PCI Ethernet Adapter #2
P4	4	QLogic 1Gb PCI Ethernet Adapter #4
VRTETHPTP		IBM iSeries Virtual Ethernet Point-to-Point
VRTETH0		IBM iSeries Virtual Ethernet 0
VRTETH1		IBM iSeries Virtual Ethernet 1
VRTETH2		IBM iSeries Virtual Ethernet 2
VRTETH3		IBM iSeries Virtual Ethernet 3

Figure 9-44 QVNI MAP /inrland switch

Read Figure 9-44 as follows:

► **Port**

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-41 on page 410 to determine the SCSI MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

VRTETHxxx is the short name for a Virtual Ethernet LAN. Note that VRTETHPTP is a stand-alone point-to-point Virtual Ethernet LAN between the hosted server and hosting i5/OS partition.

► **i5/OS RMTIFC**

Numbers 1 - 4 correspond to the numbers in the Interfaces column under the Network Interfaces tab of the RMTSYS Properties. They often correspond to P1 - P4, but not necessarily.

► **Description**

This is descriptive information taken directly from the Windows Network Connections display.

In this example, the first four lines show descriptive information for the LAN side of the initiator HBA ports. The last five lines show descriptive information for the Virtual Ethernet LAN adapters.

/inrlannam

The /inrlannam switch displays descriptive information for each Virtual Ethernet LAN network connection in Windows. An example is shown in Figure 9-45.

Initiator LAN Details -- Connection Name table:		
Port	i5/OS RMTIFC	Connection Name

P1	1	Local Area Connection
P2	2	Local Area Connection 4
P3	3	Local Area Connection 3
P4	4	Local Area Connection 2
VRTETHPTP		Local Area Connection 5
VRTETH0		Local Area Connection 6
VRTETH1		Local Area Connection 7
VRTETH2		Local Area Connection 8
VRTETH3		Local Area Connection 9

Figure 9-45 QVNIMAP /inrlannam switch

Read Figure 9-45 as follows:

► **Port**

P1 - P4 represent physical initiator HBA ports. You can tie P1 - P4 back to a physical initiator HBA by using Figure 9-43 on page 412 to determine the LAN MAC address for P1 - P4. The MAC addresses for an initiator HBA port are printed on a sticker attached to the adapter.

VRTETHxxx is the short name for a Virtual Ethernet LAN. Note that VRTETHPTP is a stand-alone point-to-point Virtual Ethernet LAN between the hosted server and hosting i5/OS partition.

► **i5/OS RMTIFC**

Numbers 1 - 4 correspond to the numbers in the Interfaces column under the Network Interfaces tab of the RMTSYS Properties. They often correspond to P1 - P4, but not necessarily.

► **Connection Name**

This is descriptive information taken directly from the Windows Network Connections display.

In this example, the first four lines show descriptive information for the LAN side of the initiator HBA ports. The last five lines show descriptive information for the Virtual Ethernet LAN adapters.

/tgtstgadr

The /tgtstgadr switch displays IP address information for the SCSI side of each target HBA. An example is shown in Figure 9-46.

Target SCSI Details -- Address table:		
NWSH	Path	IP Address
HNWSH4	1	192.168.99.204
HNWSH5	2	192.168.99.205
HNWSH6	3	192.168.99.206
HNWSH7	4	192.168.99.207

Figure 9-46 QVNIMAP /tgtstgadr switch

Read Figure 9-46 as follows:

► **NWSH**

Each NWSH represents a target HBA.

You can tie an NWSH back to a physical target HBA as follows:

- Determine the HBA's hardware resource ID specified in the NWSH (LINxx).
- Run the CL command WRKHDWRSC *CMN in a green screen.
- Enter a 7 against the LINxx resource in WRKHDWRSC *CMN.
- The Frame ID and Card position enable you to locate the HBA in a tower.

► **Path**

This is a distinguishing number for each storage path that has been created for an NWSH. This number corresponds to the number listed in the Path column under the Storage Paths tab of the hosted server's NWSD Properties.

► **IP Address**

The IP addresses shown here belong to the SCSI side of the target HBAs in the hosting partition. Note that the SCSI and LAN sides of the iSCSI HBA ports each have their own unique IP addresses. For more information about IP addressing in a System i iSCSI network, refer to 9.12, "IP addressing in a System i iSCSI network" on page 401.

/tgtstgiqn

The /tgtstgiqn switch displays the IQNs for the SCSI side of each target HBA. An example is shown in Figure 9-47 on page 416.

Target SCSI Details -- IQN table:

NWSH	Path	iSCSI Qualified Name (IQN)
HNWSH4	1	iqn.1924-02.com.ibm:10d8c1f2.etnwsd11.t1
HNWSH5	2	iqn.1924-02.com.ibm:10d8c1f2.etnwsd11.t2
HNWSH6	3	iqn.1924-02.com.ibm:10d8c1f2.etnwsd11.t3
HNWSH7	4	iqn.1924-02.com.ibm:10d8c1f2.etnwsd11.t4

Figure 9-47 QVNIMAP /tgtstgiqn switch

Read Figure 9-47 as follows:

► **NWSH**

Each NWSH represents a target HBA.

You can tie an NWSH back to a physical target HBA as follows:

- Determine the HBA's hardware resource ID specified in the NWSH (LINxx).
- Run the CL command WRKHDWRSC *CMN in a green screen.
- Enter a 7 against the LINxx resource in WRKHDWRSC *CMN.
- The Frame ID and Card position enable you to locate the HBA in a tower.

► **Path**

This is a distinguishing number for each storage path that has been created for an NWSH. This number corresponds to the number listed in the Path column under the Storage Paths tab of the hosted server's NWSD Properties.

► **iSCSI Qualified Name (IQN)**

Each *target HBA* SCSI port not only has a unique MAC and IP address, but it also has a unique IQN. In the wider storage context, an IQN is an industry standard term that uniquely represents a storage area network (SAN) interface.

Unlike initiator IQNs, target IQNs cannot be user-defined. Therefore, target IQNs are for display only.

For more information about IQNs, refer to the following Web site:

<http://www.ietf.org/rfc/rfc3720.txt>

The *target HBA* SCSI port IQN is comprised of several parts. For example, the IQN in Figure 9-29 on page 381 for NWSH4 is:

iqn.1924-02.com.ibm:10d8c1f2.etnwsd11.t1

The components of this IQN and their meanings are as follows:

– **iqn**

The iqn prefix designates this string as an industry standard IQN.

– **1924-02**

The date code 1924-02 is based on the year and month when IBM starting using IBM as the company name.

– **.com.ibm**

.com.ibm is equivalent to a domain suffix for a storage area network (SAN).

– **10d8c1f2**

This string is the i5/OS hosting partition serial number (with any letters in lower case).

– **etnwsd11**

This string is the NWSD name for the hosted server (with any letters in lower case).

– **t1**

These characters represent “t” for *target* followed by a character that distinguishes this IQN from all others in this hosting partition. Note that future enhancements to the implementation of iSCSI on System i might allow the last character to be a letter rather than a number.

/tgtlanadr

The /tgtlanadr switch displays IP address information for the LAN side of each target HBA. An example is shown in Figure 9-48.

Target LAN Details -- Address table:	
NWSH	IP Address
-----	-----
HNWSH4	192.168.99.224
HNWSH5	192.168.99.225
HNWSH6	192.168.99.226
HNWSH7	192.168.99.227

Figure 9-48 QVNiMAP /tgtlanadr switch

Read Figure 9-48 as follows:

► **NWSH**

Each NWSH represents a target HBA.

You can tie an NWSH back to a physical target HBA as follows:

- Determine the HBA’s hardware resource ID specified in the NWSH (LINxx).
- Run the CL command WRKHDWRSC *CMN in a green screen.
- Enter a 7 against the LINxx resource in WRKHDWRSC *CMN.
- The Frame ID and Card position enable you to locate the HBA in a tower.

► **IP Address**

The IP addresses shown here belong to the LAN side of the target HBAs in the hosting partition. Note that the SCSI and LAN sides of the iSCSI HBA ports each have their own unique IP addresses. For more information about IP addressing in a System i iSCSI network, refer to 9.12, “IP addressing in a System i iSCSI network” on page 401.

/csv

The /csv switch generates output in a .CSV format. For example:

```
C:\>qvnimap /stg /csv > c:\output2.csv
```

This example redirects all iSCSI storage-related information to the file output2.csv in comma-separated value format.

9.14 Scaling tasks

To set up the configurations described in this chapter, there are certain tasks that you need to complete. Once you have chosen the configuration you want to set up from either 9.5, “Scenarios for scaling a hosted xSeries server connection” on page 343 or 9.6, “Scenarios for

scaling a hosted Blade server connection” on page 352, go back and print off the list of tasks under the Setup heading for that configuration. Then return to this section and work through the tasks on your list.

Important: This section, and indeed this whole chapter, assumes that you have previously set up a basic iSCSI configuration for either an xSeries or Blade server. If you have not, then you need to perform a basic iSCSI configuration before you can proceed. To set up a basic iSCSI configuration, work through the instructions on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>

The tasks required to set up the configurations described in 9.5, “Scenarios for scaling a hosted xSeries server connection” on page 343 and 9.6, “Scenarios for scaling a hosted Blade server connection” on page 352 are listed below. Select each task you need to complete and you are referred to the relevant setup steps, either in this chapter or on the Web:

► **Install an additional target HBA in the hosting partition**

Refer to 9.14.1, “Installing an additional target HBA in the hosting partition” on page 419.

► **Configure an additional target HBA in the hosting partition**

Refer to 9.14.2, “Configuring an additional target HBA in the hosting partition” on page 419.

► **Create a storage path using the additional target HBA**

Refer to 9.14.3, “Creating a storage path using the additional target HBA” on page 423.

► **Install an additional initiator HBA in an xSeries server**

Refer to 9.14.4, “Installing an additional initiator HBA in an xSeries server” on page 425.

► **Configure an additional initiator HBA port in an xSeries server**

Refer to 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426.

This task is composed of subtasks:

- a. First, you must configure the additional initiator HBA port as a non-boot port *or* a boot port:

- **Configure the additional initiator HBA port as a non-boot port**

Refer to 9.14.6, “Configuring the additional initiator HBA port as a non-boot port” on page 426.

- **Configure the additional initiator HBA port as the boot port**

Refer to 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429.

- b. Secondly, you must update the RMTSYS configuration object with the boot port and *iSCSI* IP addressing information:

- **Update the RMTSYS configuration object**

Refer to 9.14.8, “Updating the RMTSYS configuration object” on page 433.

► **Configure the second initiator HBA port in a Blade server**

Refer to 9.14.9, “Configuring the second initiator HBA port in a Blade server” on page 436.

9.14.1 Installing an additional target HBA in the hosting partition

The #5783 and #5784 iSCSI target HBAs are customer install features.

To install one of these adapters you need to determine a valid slot for the adapter and then follow the normal rules for installing an adapter in the System i. Note that the target HBA does not require an IOP, so slot placement is very flexible. You can install a target HBA in most slots in the System i system unit and I/O towers.

Once the adapter has been installed, you must cable the adapter into the iSCSI network.

If you need additional information, refer to the following document:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphak/iphak.pdf>

The document is called *System p5 and i5: eServer p5 and i5 and OpenPower PCI adapters*. Open the following topics: **PCI adapter** → **PCI adapter placement for IBM System i and eServer i5 system units and expansion units**.

If you are still having difficulty, place a call with IBM hardware support.

9.14.2 Configuring an additional target HBA in the hosting partition

Once the iSCSI adapter has been physically installed in the System i, you need to configure it so that storage spaces and Virtual Ethernet LANs can use it. When you configure a target HBA, you create an object called a network server host adapter (NWSH).

Using iSeries Navigator

To configure a target HBA using iSeries Navigator, follow these steps:

1. Expand **Configuration and Service** → **Hardware** → **All Hardware** as shown in Figure 9-49 on page 420. Record the Resource ID, Lin06 in this example.

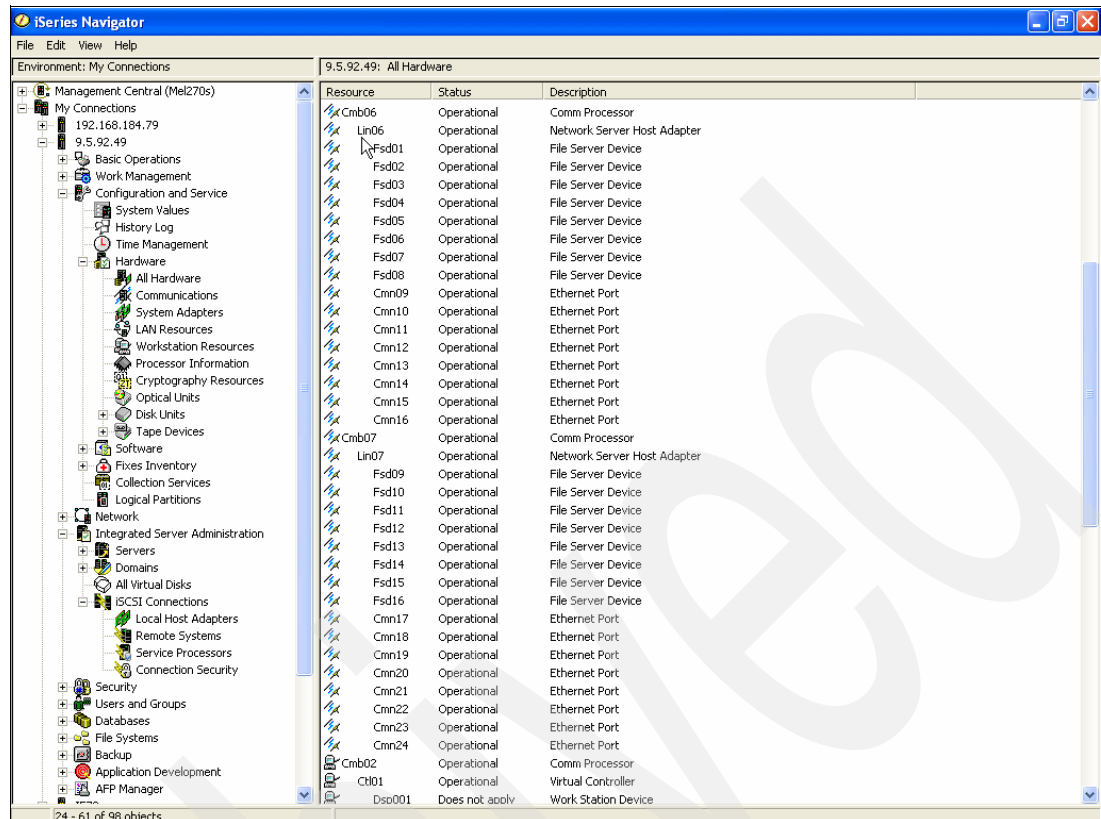


Figure 9-49 Identifying the iSCSI target HBA Resource ID - iSeries Navigator

2. Expand **Integrated Server Administration** → **iSCSI Connections** → **Local Host Adapters**.

3. Right-click **Local Host Adapters** and select **New Network Server Host Adapter**.

Fill in the parameters using the examples shown in Figure 9-50 and Figure 9-51 on page 421 as a guide. Note that in Figure 9-51 on page 421, we are using the recommended IP addressing schema described in 9.12.2, "Choosing an IP addressing schema" on page 404.

Note also that you must fill in a gateway IP address although it is not used.

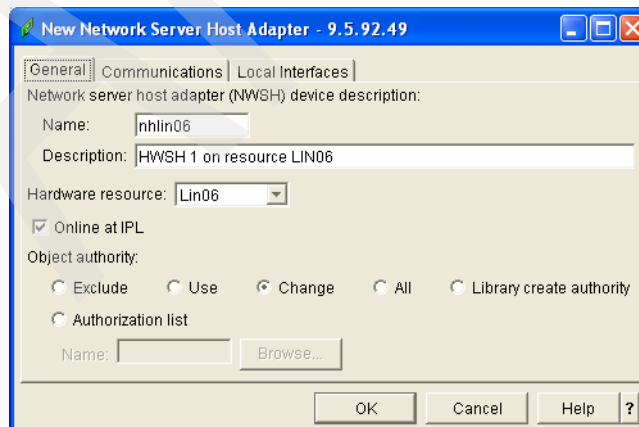


Figure 9-50 Creating a network server host adapter - General tab

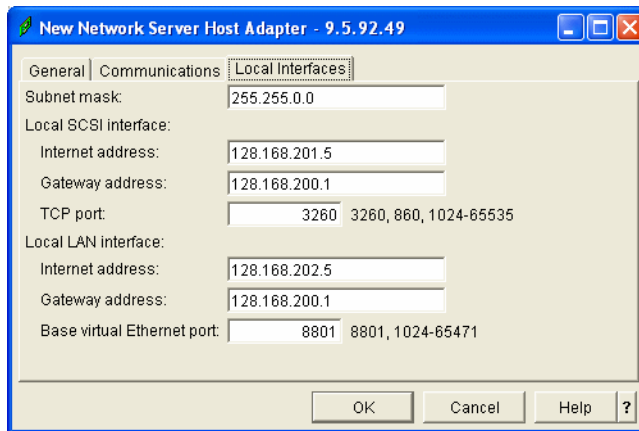


Figure 9-51 Creating a network server host adapter - Local interface tab

Note that under the Communications tab, you can enter an optional message queue for communications messages. Otherwise, they go to QSYSOPR.

4. Click **OK** to create the object.
5. Activate the network server host adapter by right-clicking on it and selecting **Start**. The status should go to **Active**.

The network server host adapter is now ready for use.

Using green screen

To configure a target HBA using green screen, follow these steps:

1. Obtain the hardware resource ID for the adapter.

From a green screen command line, type `WRKHDWRSC *CMN`. Press Enter. You see a display similar to the one shown in Figure 9-52 on page 422.

Work with Communication Resources					System: ITCDEM02
Type options, press Enter.					
5=Work with configuration descriptions 7=Display resource detail					
Opt	Resource	Type	Status	Text	
	CMB06	573B	Operational	Comm Processor	
	LIN06	573B	Operational	Network Server Host Adapter	
	FSD01	573B	Operational	File Server Device	
	FSD02	573B	Operational	File Server Device	
	FSD03	573B	Operational	File Server Device	
	FSD04	573B	Operational	File Server Device	
	FSD05	573B	Operational	File Server Device	
	FSD06	573B	Operational	File Server Device	
	FSD07	573B	Operational	File Server Device	
	FSD08	573B	Operational	File Server Device	
	CMN09	6B01	Operational	Ethernet Port	
	CMN10	6B01	Operational	Ethernet Port	
	CMN11	6B01	Operational	Ethernet Port	
	CMN12	6B01	Operational	Ethernet Port	
	CMN13	6B01	Operational	Ethernet Port	
					More...
F3=Exit F5=Refresh F6=Print F12=Cancel					

Figure 9-52 Identifying the target HBA Resource ID - green screen

- Look for a Network Server Host Adapter resource of type 573B (copper) or 573C (fiber). Record the Resource ID, LIN06 in this example.
- Type the command CRTDEVNWSH and press F4.

Fill in the parameters using the example shown in Figure 9-53 on page 423 as a guide. Note that in Figure 9-53 on page 423 we are using the recommended IP addressing schema described in 9.12.2, "Choosing an IP addressing schema" on page 404.

Note also that you must fill in a gateway IP address although it is not used.

- You must fill in a gateway IP address although it is not used.
- We strongly recommend that you change Online at IPL to *YES.
- When you page down you can enter an optional message queue for communications messages. Otherwise they go to QSYSOPR.
- Leave the Recovery limits as the defaults unless you have a specific reason to change them.


```

Create Device Desc (NWSH) (CRTDEVNWSH)

Type choices, press Enter.

Device description . . . . . > NHLIN06      Name
Resource name . . . . . > LIN06           Name
Local interface:
  Subnet mask . . . . . 255.255.0.0
  Port speed . . . . . *AUTO             *AUTO
  Duplex . . . . . *AUTO                 *AUTO
Local SCSI interface:
  Internet address . . . . . 128.168.201.5
  Gateway address . . . . . 128.168.200.1
  SCSI TCP port . . . . . 3260           1024-65535, 3260, 860
Local LAN interface:
  Internet address . . . . . 128.168.202.5
  Gateway address . . . . . 126.168.200.1
  Virtual Ethernet base UDP port 8801     1024-65471, 8801
  Online at IPL . . . . . *YES           *NO, *YES

More...

F3=Exit  F4=Prompt  F5=Refresh  F10=Additional parameters  F12=Cancel
F13=How to use this display  F24=More keys

```

Figure 9-53 Creating a Network server host adapter - green screen

Page down and enter a name for the device.

4. Press Enter to create the object.
5. Vary on the network server host adapter by typing the command `WRKCFGSTS *DEV *NWSH`. Enter option 1 against the object. The status should go to ACTIVE.

The network server host adapter is now ready for use.

9.14.3 Creating a storage path using the additional target HBA

In order to deploy storage spaces on an additional target HBA that you have installed in the System i, you need to create a storage path for each hosted server using the network server host adapter (NWSH) that you have created for the additional target HBA. You can then use the storage path to link storage spaces to the NWSD for each hosted server. Using a single NWSH you can create storage paths for up to eight hosted servers using each hosted server's NWSD. Linking storage spaces to an NWSD using a storage path is called path deployment, and is described in 9.15, "Path deployment tasks" on page 436.

Note: Note that you do not need to create a storage path in order to deploy a Virtual Ethernet LAN to a target HBA. To deploy or redeploy a Virtual Ethernet LAN to a new target HBA, refer to 9.15.2, "Redeploying a Virtual Ethernet LAN to a different target HBA" on page 441.

Using iSeries Navigator

To create a storage path for a hosted server using the additional target HBA and iSeries Navigator, follow these steps:

1. Expand **Integrated Server Administration**.
2. Click **Servers**.
3. Right-click the hosted server you want to add the storage path to. Select **Properties**.
4. Select the **Storage paths** tab as shown in Figure 9-54.

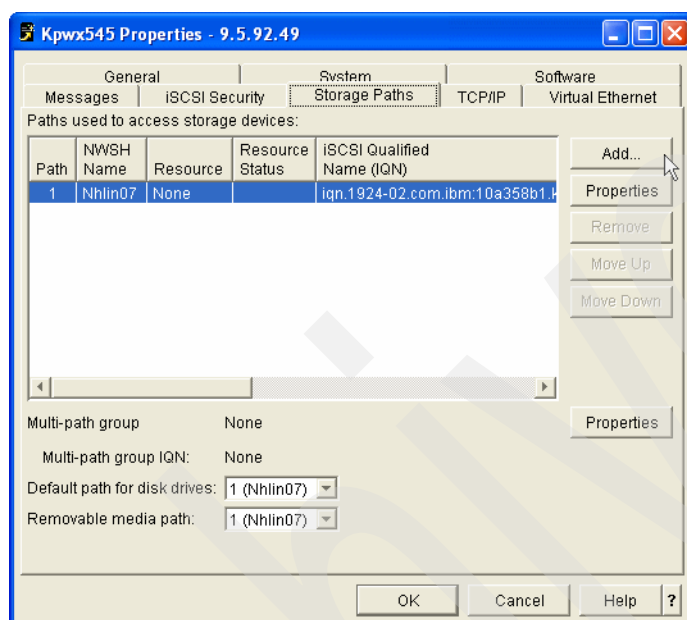


Figure 9-54 Creating a storage path for a hosted server - 1

5. Click **Add**. You see a window similar to the one displayed in Figure 9-55. Select the network server host adapter (NWSH) you want to use to create the storage path from the drop-down list. Click **OK**.

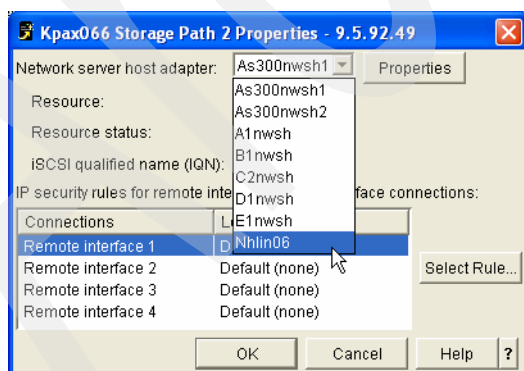


Figure 9-55 Creating a storage path for a hosted server - 2

6. The network server host adapter is added to the list of available storage paths for this hosted server as shown in Figure 9-56 on page 425. Note that the new storage path does not display an IQN until the hosted server has been restarted and a connection has been made between the new storage path and an initiator HBA port.

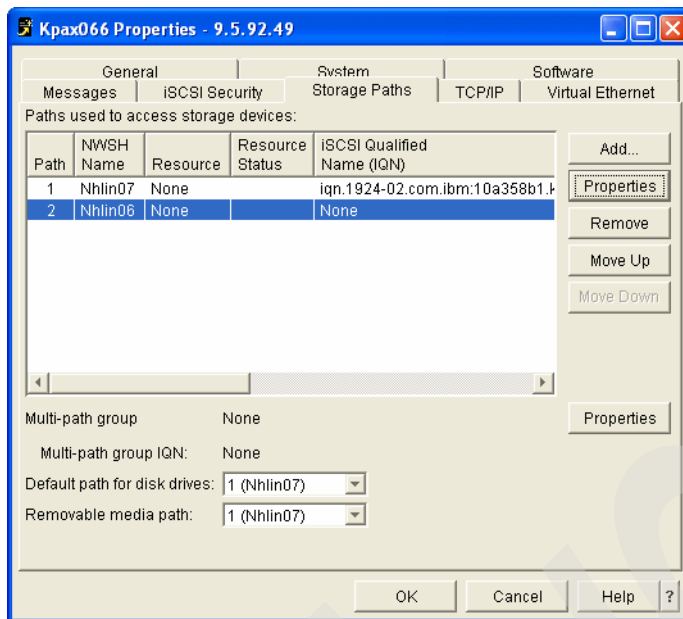


Figure 9-56 Creating a storage path for a hosted server - 3

7. Click **OK** on the Properties window to save the new storage path in the NWSD.
8. Start the hosted server.

Your new storage path is now ready to be used, or more specifically, used to link storage spaces to the NWSD. In order for storage spaces to use the new storage path, you need to manually deploy them to this storage path as described in 9.9, “Manual path deployment - target side” on page 377. Note that storage spaces and Virtual Ethernet LANs are never automatically deployed on the target side.

9.14.4 Installing an additional initiator HBA in an xSeries server

Important: You cannot install an additional initiator HBA adapter in a Blade server because Blade servers are restricted to one initiator HBA adapter only. However, the initiator HBA adapter in a Blade server does have a second port that you can configure as described in 9.14.9, “Configuring the second initiator HBA port in a Blade server” on page 436.

To install an initiator HBA adapter in an xSeries server, follow these steps:

1. Using the documentation that comes with the adapter, install it in a supported slot.
2. Update the adapter firmware by downloading the latest version from the Web and installing it on the xSeries server. Note that you must update the adapter firmware each time you install an initiator HBA adapter in the xSeries server.
3. Cable the adapter into the iSCSI network.

If you need additional information, refer to the following document:

<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphcz/iphcz.pdf>

The document is called *System i and System p: iSCSI Host Bus Adapter for IBM BladeCenter or xSeries*. Start at the beginning and work through the document.

If you are still having difficulty, place a call with IBM hardware support.

9.14.5 Configuring an additional initiator HBA port in an xSeries server

After you install an additional initiator HBA adapter in an xSeries server, you need to configure it before it can be used. There are two basic steps to this procedure:

1. Configuring the additional initiator HBA port firmware settings

The additional initiator HBA port firmware settings need to be configured using the CTRL-Q utility in one of two ways:

– Configuring the additional initiator HBA port as a non-boot port

We are assuming here that you have already set up a basic iSCSI configuration and, therefore, the boot port is already configured. Given that you can only have one boot port, you would normally configure an additional initiator HBA port as a non-boot port. The configuration of a non-boot initiator HBA port is different than a boot port.

Configuring a non-boot port is described in 9.14.6, “Configuring the additional initiator HBA port as a non-boot port” on page 426.

– Configuring the additional initiator HBA port as the boot port

Only one initiator HBA port in the hosted server can be configured as the boot port. As mentioned previously, the boot port should already be configured. Therefore, you would not normally need to configure the additional initiator HBA port as the boot port. However, there might be situations where you need to change the boot port configuration, or configure it on another initiator HBA port. We therefore describe how to configure the initiator boot port here for completeness.

Configuring the boot port is described in 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429.

2. Configuring the additional initiator HBA port settings in the RMTSYS configuration object

In order for i5/OS to connect to an initiator HBA port, the port settings need to be configured in the remote system configuration object (RMTSYS) for the hosted server. This procedure is described in 9.14.8, “Updating the RMTSYS configuration object” on page 433.

9.14.6 Configuring the additional initiator HBA port as a non-boot port

To configure an additional initiator HBA port as a *non-boot* port from either the server’s local console, or from a Remote Control session on the Service Processor (or Management Module), follow these steps:

1. Power on the xSeries or Blade server.
2. While the server is powering up, press CTRL-Q when you see the following line on the display:

Press <CTRL-Q for Fast!UTIL>

3. After a minute or two, you see the display shown in Figure 9-57 on page 427.

Note that in this example, configuration was performed remotely using a Remote Control session on the Service Processor of an xSeries server.

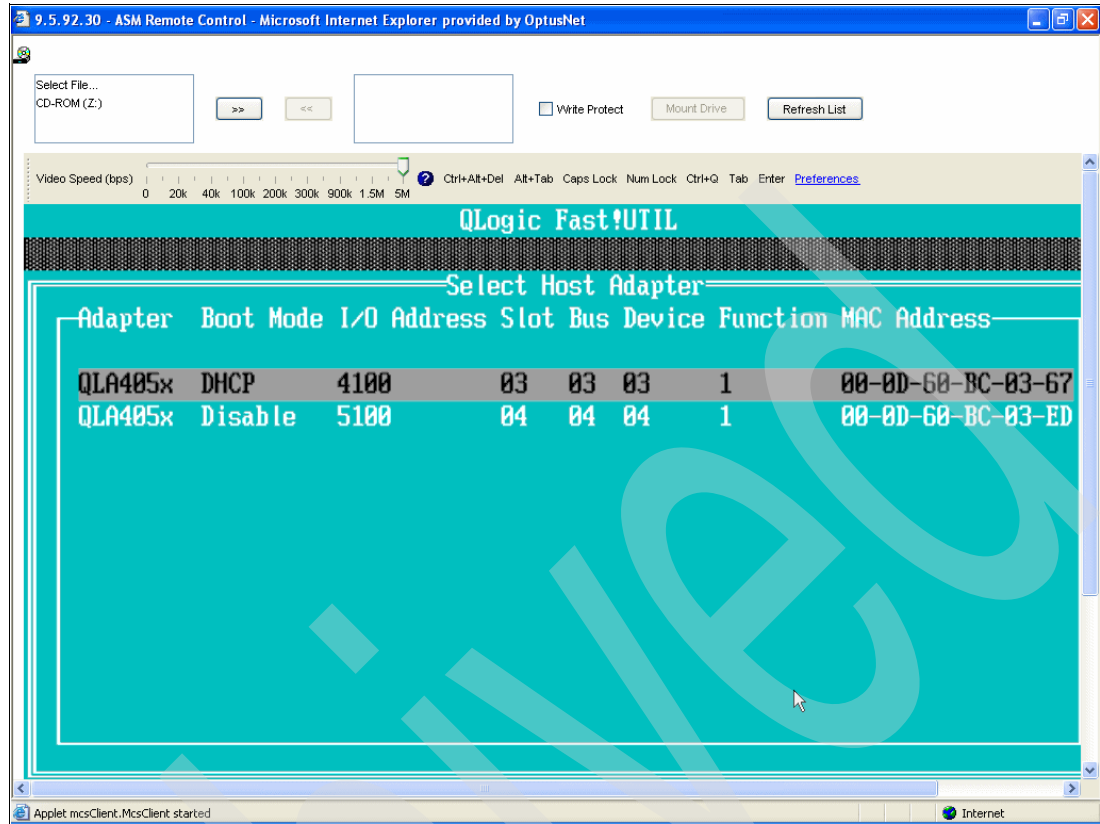


Figure 9-57 QLogic Fast!UTIL display

Figure 9-57 shows two initiator HBA ports that are installed in the xSeries or Blade server, their MAC addresses, and other configuration information.

In Figure 9-57, the second port is the one, which has just been installed. You can tell which port has just been installed from its MAC address, because you would have previously recorded the MAC address of the first port you installed in order to configure the IP addresses in the RMTSYS object for this xSeries or Blade server.

Alternatively, you can tell which port has just been installed by comparing the MAC addresses shown in Figure 9-57 with the MAC addresses printed on a sticker attached to the adapter. An xSeries HBA has a label with two MAC addresses (one for the SCSI side and the other for the LAN side), whereas a Blade HBA has a label with four MAC addresses (one for the SCSI side and the other for the LAN side of both ports). The SCSI side MAC is labelled "iSCSI" and the LAN side is labelled "TOE" (TCP Offload Engine). Note that you need to record the MAC addresses on the Blade HBA before you install it, because the sticker is not visible once the HBA is installed.

4. Record the following information from this display:
 - Identify the *first* initiator HBA port you set up. This is the port that the Windows server boots from because its Boot Mode is set to DHCP. Write down the following information for the *boot port*:
 - Bus:
 - Device:
 - Function:

In the example shown in Figure 9-57, these values are: 3, 3, and 1 respectively. You need this information later when you specify the boot port in the RMTSYS object.

- Record the MAC address of the initiator HBA port you just installed, *not* the boot port! In Figure 9-57 on page 427, the value is: 00 0D 60 BC 03 ED.
- 5. Highlight the initiator HBA port you just installed, *not* the boot port, by pressing the down arrow (if necessary) and then Enter. You are now going to configure the additional initiator HBA port as a *non-boot port*.
- 6. The Fast!UTIL Options menu is displayed. From now on, any settings you change are changed only for the port you selected on the previous display.
Select **Configuration Settings**. Press Enter.
- 7. The Configuration Settings menu shown in Figure 9-58 appears. You navigate your way around this utility by using the Up and Down arrow keys to select a menu option, then press Enter. To back up, press the Escape key.

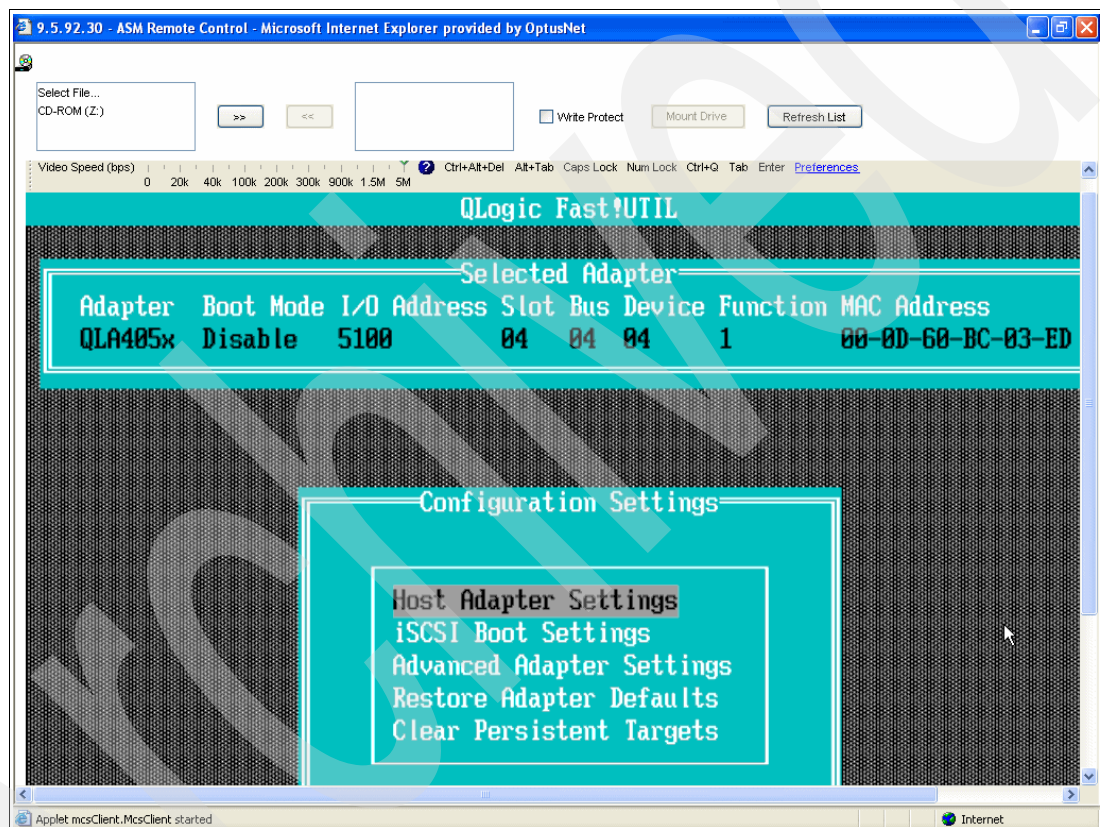


Figure 9-58 The Configuration Settings menu

- 8. From the Configuration Settings menu, select **Clear Persistent Targets**. Press Enter. Clear the Persistent Targets and then Escape back to the Configuration Settings menu.
- 9. From the Configuration Settings menu, select **Host Adapter Settings**. Press Enter.
 - a. Select **LUNs per Target**. Press Enter.
Change the setting from 16 up to 64, if required. 64 is the maximum number of System i storage spaces that can be accessed by this port. Do not set the value higher than you need.
 - b. Select **Initiator IP Address via DHCP**. Press Enter to change the setting to **No**.
 - c. Escape back to the Configuration Settings menu.
- 10. From the Configuration Settings menu, select **iSCSI Boot Settings**. Press Enter.

- a. Select **Adapter Boot Mode**. Press Enter to disable it if it is not already disabled.
 - b. Select **Primary Boot Device Settings**. Press Enter.
 - c. Select **Security Settings**. Press Enter.
 - d. Select **Chap**. Press Enter to disable it if it is not already disabled.
11. Escape back to the Configuration Settings menu, or until you are prompted to **Save Changes**. Press Enter to save your changes. This takes a couple of minutes.
 12. From the Configuration Settings menu, select **Advanced Adapter Settings**. Press Enter.
 - a. Select **Delayed ACK**. Press Enter to disable it.
 - b. Select **MTU**. Press Enter to change it to 1500 or 9000.

We recommend that you set the MTU size to 1500 because testing has shown that this setting provides best performance in most cases, even if the switch supports 9000. Note also that not all switches support the larger MTU size.
 - c. Escape back to the Configuration Settings menu.
 - d. Escape back to the Fast!UTIL Options menu.
 13. Press Escape again. You are prompted to **Save changes**. Press Enter. This takes a couple of minutes.
 14. From the Fast!UTIL Options menu select **Exit Fast!UTIL**. Press Enter.
 15. You are prompted to **Reboot System**. Press Enter. After the xSeries or Blade server has rebooted, power it off.
 16. Assuming that you have already installed Windows on this server, start it up from iSeries Navigator.
 17. After you log on to Windows from the local console or Remote Console display, you should see Windows plug and play the new iSCSI initiator HBA port. Check in Device Manager that it is correctly configured. Note that if you are setting up the second initiator port on a Blade server, you do not see Windows plug and play because the second port is on the same physical daughter board as the first port, and, therefore, Windows uses the same driver for both ports.

You now need to specify the boot port and configure the iSCSI network interfaces in the RMTSYS object so that i5/OS can establish connections to the new initiator HBA port. Proceed to 9.14.8, "Updating the RMTSYS configuration object" on page 433 for a description of how to do this.

9.14.7 Configuring the additional initiator HBA port as the boot port

To configure an additional initiator HBA port as a *boot port* in an xSeries or Blade server from either server's local console, or from a Remote Control session on the Service Processor (or Management Module), follow these steps:

1. Power on the xSeries or Blade server.
2. While the server is powering up, press CTRL-Q when you see the following line on the display:

Press <CTRL-Q for Fast!UTIL>
3. After a minute or two, you see the display shown in Figure 9-59 on page 430.

Note that in this example, configuration was performed remotely using a Remote Control session on the Service Processor of an xSeries server.

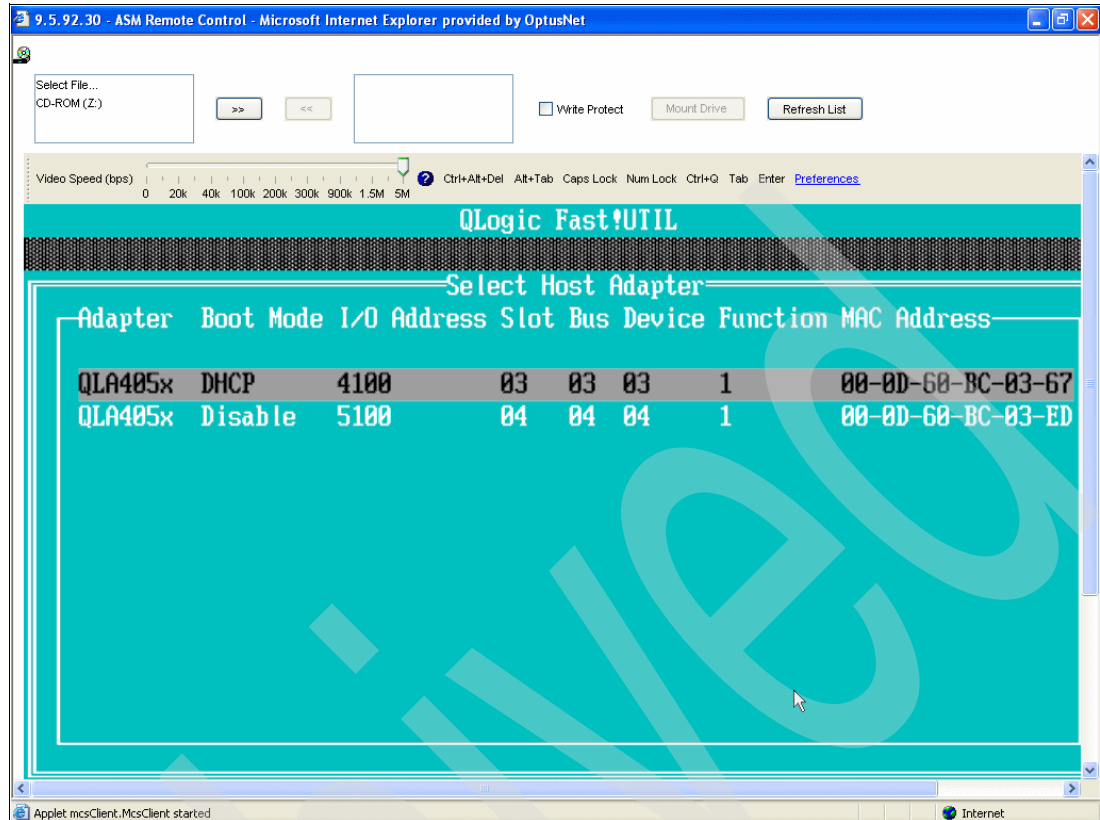


Figure 9-59 QLogic Fast!UTIL display

Figure 9-59 shows two initiator HBA ports that are installed in the xSeries or Blade server, their MAC addresses and other configuration information.

In Figure 9-59, the second port is the one which has just been installed. You can tell which port has just been installed from its MAC address, because you would have previously recorded the MAC address of the first port you installed in order to configure the IP addresses in the RMTSYS object for this xSeries or Blade server.

Alternatively, you can tell which port has just been installed by comparing the MAC addresses shown in Figure 9-59 with the MAC addresses printed on a sticker attached to the adapter. An xSeries HBA has a label with two MAC addresses (one for the SCSI side and the other for the LAN side) whereas a Blade HBA has a label with four MAC addresses (one for the SCSI side and the other for the LAN side of both ports). The SCSI side MAC is labelled "iSCSI" and the LAN side is labelled "TOE" (TCP Offload Engine). Note that you need to record the MAC addresses on the Blade HBA before you install it because the sticker is not visible once the HBA is installed.

4. Record the following information from this display:

- Identify the *first* initiator HBA port you set up. This is the port that the Windows server boots from because its Boot Mode is set to DHCP.

Here we assume that you want to set up the initiator HBA port you have just installed as the boot port. In other words, you are *changing* the boot port from the port with MAC address 00 0D 60 BC 03 67 to the port with MAC address 00 0D 60 BC 03 ED.

Write down the following information for the initiator HBA port you just installed, which is going to be your *new* boot port:

- Bus:
- Device:

- Function:

In the example shown in Figure 9-59 on page 430, these values are: 4, 4, and 1 respectively. You need this information later when you specify the boot port in the RMTSYS object.

- Record the MAC address of the initiator HBA port you just installed, which is going to be your new boot port. In Figure 9-59 on page 430, the value is: 00 0D 60 BC 03 ED.
- Highlight the initiator HBA port you just installed by pressing the down arrow (if necessary), and then Enter. You are now going to configure the additional initiator HBA port as a *boot port*.
 - The Fast!UTIL Options menu is displayed. From now on, any settings you change are changed only for the initiator HBA port you selected on the previous display.
Select **Configuration Settings**. Press Enter.
 - The Configuration Settings menu shown in Figure 9-60 appears. You navigate your way around this utility by using the Up and Down arrow keys to select a menu option, then press Enter. To back up, press the Escape key.

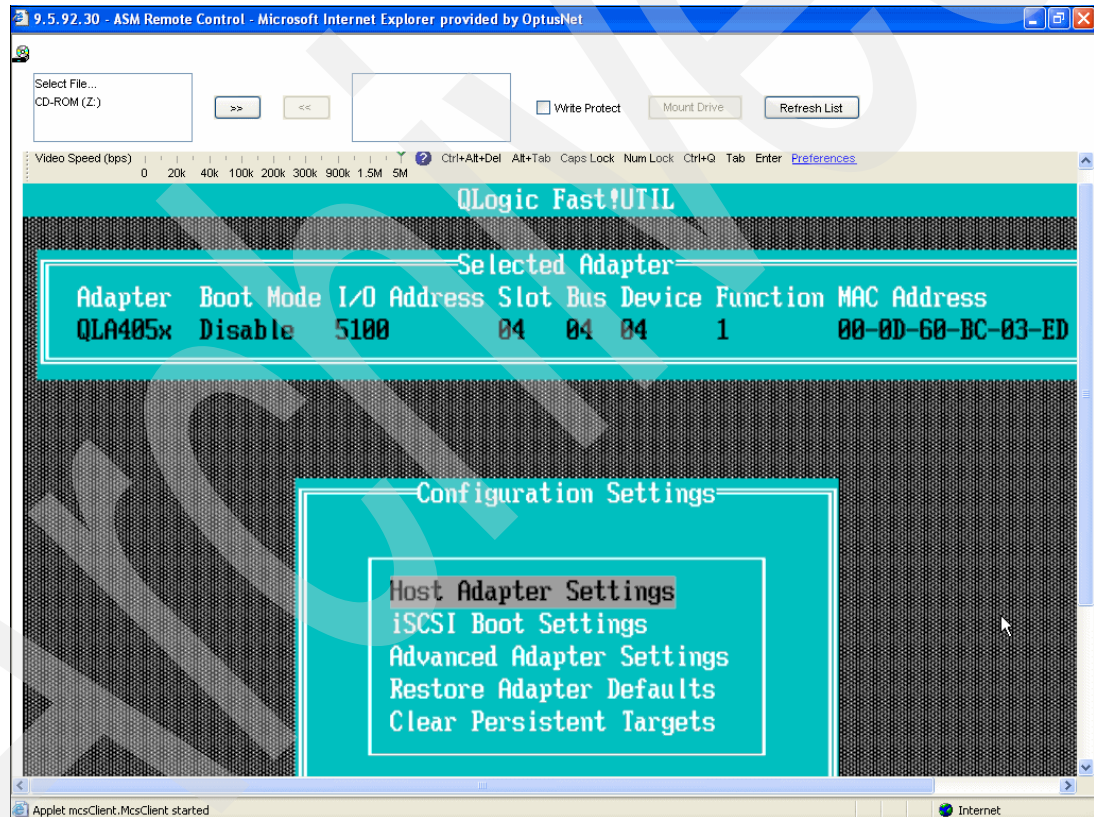


Figure 9-60 The Configuration Settings menu

- From the Configuration Settings menu, select **Clear Persistent Targets**. Press Enter.
Clear the Persistent Targets and then Escape back to the Configuration Settings menu.
- From the Configuration Settings menu, select **Host Adapter Settings**. Press Enter.
 - Select **LUNs per Target**. Press Enter.
Change the setting from 16 up to 64, if required. 64 is the maximum number of System i storage spaces that can be accessed by this port. Do not set the value higher than you need.

- b. Select **Initiator IP Address via DHCP**. Press Enter to change to **No**.
 - c. Escape back to the Configuration Settings menu.
10. From the Configuration Settings menu, select **iSCSI Boot Settings**. Press Enter.
 - a. Select **Adapter Boot Mode**. Press Enter and change it to **DHCP**.
 - b. Select **Primary Boot Device Settings**. Press Enter.
 - c. Select **Security Settings**. Press Enter.
 - d. Select **Chap**. Press Enter to enable it. Enabling Chap is optional, although we recommend it. Chap is an authentication protocol that ensures that the target HBA is talking to the correct initiator HBA port and not an imposter.
 - i. Select **Chap Name** and enter a value.
 - ii. Select **Chap Secret** and enter a value.
 - iii. Re-enter the Chap Secret.
 - iv. Record the Chap Name and Chap Secret because you need to enter them in the RMTSYS object later on. Note that they are case sensitive.
11. From the Configuration Settings menu, select **Advanced Adapter Settings**. Press Enter.
 - a. Select **Delayed ACK**. Press Enter to disable it.
 - b. Select **MTU**. Press Enter to change it to 1500 or 9000 as required.

We recommend that you set the MTU size to 1500 because testing has shown that this setting provides the best performance in most cases, even if the switch supports 9000. Note also that not all switches support the larger MTU size.
 - c. Escape back to the Fast!UTIL Options menu.
12. Press Escape until you are prompted to **Save changes**. Press Enter. This takes a couple of minutes..
13. From the Fast!UTIL Options menu, select **Exit Fast!UTIL**. Press Enter.
14. You are prompted to **Reboot System**. Press Enter.

Now that you have configured the new initiator HBA port as the boot port, you need to go back and configure the previous boot port as a non-boot port.
15. While the server is powering up, press CTRL-Q when you see the following line on the display:

Press <CTRL-Q for Fast!UTIL>
16. This time, select the initiator HBA port that was previously the boot port. You now need to make it a non-boot port.

The Fast!UTIL Options menu appears. This time it has the initiator HBA port that was previously the boot port at the top of the screen. Make sure the old boot port that you want to change to a non-boot port is highlighted.

Select **Configuration Settings**. Press Enter.
17. From the Configuration Settings menu, select **Clear Persistent Targets**. Press Enter.

Clear the Persistent Targets and then Escape back to the Configuration Settings menu.
18. Using the same procedure that you used to change the settings for the new initiator HBA port, change the settings on the port that was previously the boot port to the following values:
 - Host Adapter Settings:
 - LUNs per target: 16 through to 64 as required
 - Initiator IP Address via DHCP: Yes

- iSCSI Boot Settings:
 - Adapter Boot Mode: Disable
 - Security Settings: Chap - Disabled
- Advanced Adapter Settings:
 - Delayed ACK: Disabled
 - MTU: 1500 or 9000 as required, we recommend 1500 in all cases.

The only settings you should need to change from the previous configuration are the Adapter Boot Mode and Chap.

19. Press Escape again. You are prompted to **Save changes**. Press Enter. This takes a couple of minutes.

20. Select **Exit Fast!UTIL** from the Fast!UTIL Options menu. Press Enter.

21. You are prompted to **Reboot System**. Press Enter. After the xSeries or Blade server has rebooted, power it off.

You now need to specify the boot port, and configure the iSCSI network interfaces in the RMTSYS object so that i5/OS can establish connections to the new initiator HBA port. Proceed to 9.14.8, “Updating the RMTSYS configuration object” on page 433 for a description of how to do this.

9.14.8 Updating the RMTSYS configuration object

If you have more than one initiator HBA port in your hosted server, you need to specifically tell i5/OS Integrated Server Support which initiator HBA port Windows is booting off. (If there is only one initiator HBA port in the hosted server you do not need to specify the boot port because there is no ambiguity.) You also need to enter SCSI and Virtual Ethernet LAN IP addressing information for the new port.

Using iSeries Navigator

To specify the boot port and IP addressing information in the remote system configuration object using iSeries Navigator, follow these steps:

1. Shut down the hosted server.
2. Determine the boot port information using the CTRL-Q utility as described previously. If you have been following the steps in order, you should have already recorded this information.

The information you need is:

- Bus:
- Device:
- Function:

Ensure that you are using the information recorded for the *boot* port.

3. Expand **Integrated Server Administration** → **iSCSI Connections** → **Remote Systems**.
4. Right-click the Remote system whose boot port you want to change and select **Properties**.
5. Select the **Boot Parameters** tab. You see a window similar to the one displayed in Figure 9-61 on page 434.

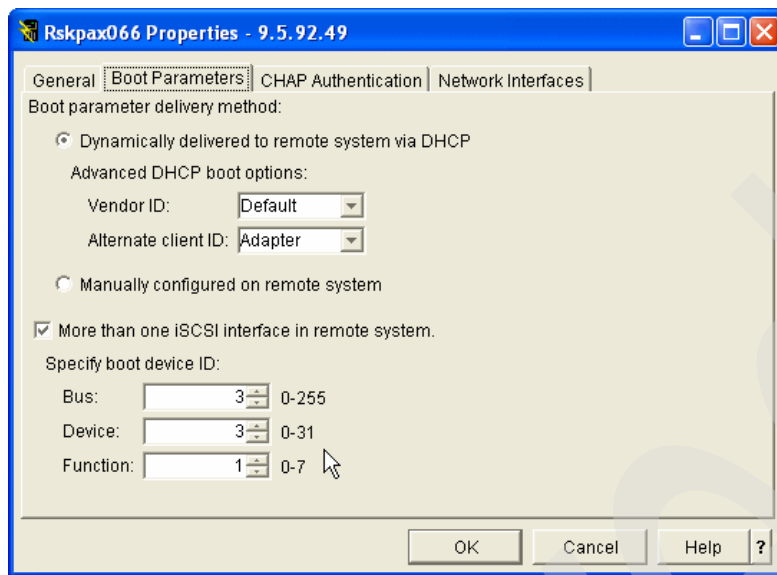


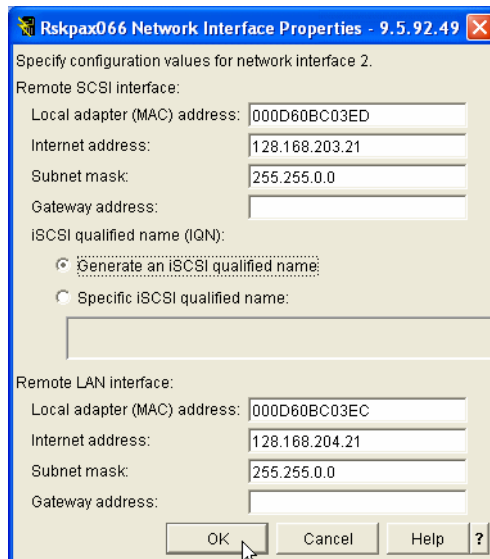
Figure 9-61 Specifying the boot device ID (boot port) in the RMTSYS configuration object

6. Check the **More than one iSCSI interface in remote system** check box if not already checked. Note that the term “iSCSI interface in remote system” is another term for an initiator HBA port.
7. Enter the values you determined for the new boot port using the CTRL-Q utility in the Bus, Device, and Function parameters.

Note: Specifying the boot device ID information in the RMTSYS configuration object is in fact optional, because these values are overridden by the values set in the CTRL-Q utility. However, a future enhancement to System i Integrated Server Support will allow you to specify values for the boot device ID through the panel shown in Figure 9-61 and override the CTRL-Q settings. You will therefore not need to run the Ctrl-Q utility to specify the boot device ID.

We recommend that you enter the CTRL-Q values in the RMTSYS configuration object as described in this section for consistency and documentation purposes.

8. Select the **Network Interfaces** tab. Click **Add** to add a new interface. You see a window similar to the one displayed in Figure 9-62 on page 435.



Rskpax066 Network Interface Properties - 9.5.92.49

Specify configuration values for network interface 2.

Remote SCSI interface:

Local adapter (MAC) address: 000D60BC03ED

Internet address: 128.168.203.21

Subnet mask: 255.255.0.0

Gateway address:

iSCSI qualified name (IQN):

☒ Generate an iSCSI qualified name:

☐ Specific iSCSI qualified name:

Remote LAN interface:

Local adapter (MAC) address: 000D60BC03EC

Internet address: 128.168.204.21

Subnet mask: 255.255.0.0

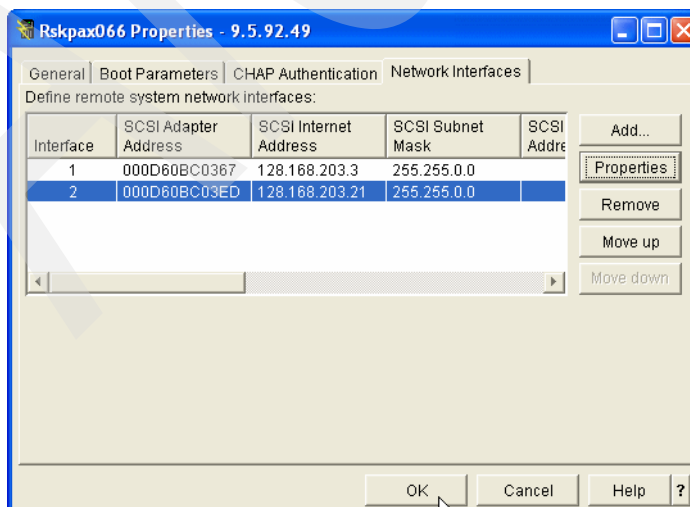
Gateway address:

OK Cancel Help ?

Figure 9-62 Adding a new remote system configuration iSCSI network interface - 1

Here you add a SCSI IP address that virtual disks can use to connect to the hosted server, and a LAN interface that Virtual Ethernet LANs can use to connect to the hosted server. Note the following points:

- The MAC address of the Remote LAN interface is always one hexadecimal digit less than the MAC address of the Remote SCSI interface. Note that the Remote SCSI interface is the one you see on the CTRL-Q display in Figure 9-59 on page 430.
 - We are using the recommended IP addressing schema described in 9.12.2, “Choosing an IP addressing schema” on page 404.
 - Make sure you select the **Generate an iSCSI qualified name (IQN)** radio button unless you want to enter a specific iSCSI qualified name. The IQN is generated the next time the initiator HBA port connects to the hosting partition.
 - Leave the Gateway address fields blank.
9. After you have finished entering iSCSI network addressing information, click **OK**. You see the second iSCSI network interface added to the list of interfaces as shown in Figure 9-63.



Rskpax066 Properties - 9.5.92.49

General | Boot Parameters | CHAP Authentication | **Network Interfaces**

Define remote system network interfaces:

Interface	SCSI Adapter Address	SCSI Internet Address	SCSI Subnet Mask	SCSI Addr
1	000D60BC0367	128.168.203.3	255.255.0.0	
2	000D60BC03ED	128.168.203.21	255.255.0.0	

Add... Properties Remove Move up Move down

OK Cancel Help ?

Figure 9-63 Adding a new remote system configuration iSCSI network interface - 2

10.If you have set up the new initiator HBA port as the *boot port*, and also changed the Chap settings from what they were set to on the old boot port, you need to update the Chap settings in the RMTSYS object using the CHAP Authentication tab. If the Chap settings have not changed, then bypass this step.

If the Chap settings in the RMTSYS object and initiator HBA port firmware settings do not match, the hosted server is not able to verify the boot drives, and the server does not start.

11.Click **OK** to save the Properties settings.

12.You can now start the hosted server.

The new iSCSI initiator HBA port is now ready to be deployed, or in other words, used by virtual disks and Virtual Ethernet LANs. You can either allow virtual disks and Virtual Ethernet LANs to be autodeployed by i5/OS Integrated Server Support, or you can manually deploy them as described in 9.10, “Manual path deployment - initiator side” on page 379.

9.14.9 Configuring the second initiator HBA port in a Blade server

You can configure the second initiator HBA port in a Blade server from either the BladeCenter’s local console, or from a Remote Control session on the Management Module. Either way, the process is exactly the same as described in 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426, except for the following points:

- ▶ You only ever see a maximum of two ports on the QLogic FastUTIL display, as shown in Figure 9-64. This is because you can only ever have one iSCSI HBA adapter with two initiator ports on a Blade server.
- ▶ Just like the xSeries server, you can configure an additional initiator HBA port in a Blade server from either the BladeCenter’s Keyboard-Video-Mouse (KVM) console, or from a Remote Control session on the Management Module. Note that the Remote Control session on a Management Module has a different appearance and works in a slightly different way than the Service Processor on an xSeries server. An example is shown in Figure 9-64.

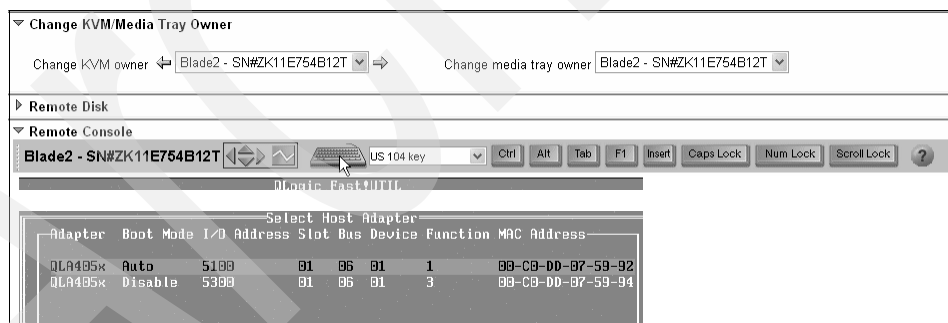


Figure 9-64 Configuring a port through the Remote Control session on the Management Module

Bearing these points in mind, return to 9.14.5, “Configuring an additional initiator HBA port in an xSeries server” on page 426 and work through those steps to configure the second initiator HBA port on the Blade server.

9.15 Path deployment tasks

In this section, we describe how to manually deploy storage spaces/virtual disks and Virtual Ethernet LANs at both the target and initiator ends. In other words, as we add additional target and initiator HBA ports to increase the bandwidth of a hosted server, we need to be

able to deploy or redeploy storage space links, virtual disk assignments, and Virtual Ethernet LAN connections to the new HBA ports to make full use of the additional bandwidth. In this section, we describe the tasks you need to perform in order to set up these connections.

In terms of deployment, there are two important aspects that you need to keep in mind:

- ▶ On the target side, storage spaces and Virtual Ethernet LANs must always be deployed or redeployed manually.
- ▶ On the initiator side, automatic deployment (and redeployment) of virtual disks and Virtual Ethernet LANs occurs by default unless you manually deploy them.

Before reading this section, you should have worked through 9.9, “Manual path deployment - target side” on page 377 and 9.10, “Manual path deployment - initiator side” on page 379 to determine the tasks you need to complete in order to deploy the storage space links, virtual disk assignments, and Virtual Ethernet LAN data paths as you would like.

Important: This section, and indeed this whole chapter, assumes that you have already set up a basic iSCSI network configuration from a hosting partition to either an xSeries or Blade server. If you have not, then you need to perform a basic iSCSI configuration before you can proceed. To set up a basic iSCSI configuration, work through the instructions on the following Web site:

<http://www.ibm.com/systems/i/bladecenter/iscsi/readme/index.html>

The tasks required to manually deploy storage spaces/virtual disks and Virtual Ethernet LANs on the target and initiator sides are listed below. Select each task you need to perform and you are referred to the relevant setup steps:

The different tasks involved in manual deployment are:

- ▶ **Redeploy a storage space to a different target HBA**
Refer to 9.15.1, “Redeploying a storage space to a different target HBA” on page 437
- ▶ **Redeploy a Virtual Ethernet LAN to a different target HBA**
Refer to 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.
- ▶ **Change the boot port**
Refer to 9.15.3, “Changing the boot port” on page 443.
- ▶ **Prevent a non-boot drive from using an initiator HBA port**
Refer to 9.15.4, “Preventing a non-boot drive from using an initiator HBA port” on page 444.
- ▶ **Redeploy a Virtual Ethernet LAN to a different initiator HBA port**
Refer to 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.

9.15.1 Redeploying a storage space to a different target HBA

To redeploy a storage space from one target HBA to another, you *unlink* the storage space from one target HBA and *relink* it to another. Note that when we redeploy a storage space from one target HBA to another we are not changing the hosted server (network server description) that the storage space is linked to, we are simply changing the data path that the storage space uses to send data to the hosted server.

Important: If you want to redeploy the boot drives to a different target HBA, be aware that you must also clear persistent data on the initiator boot port. We include this as an optional procedure at the end of this section.

Note that the boot drives (C: and D:) must always be linked to the same NWSH.

Using iSeries Navigator

To redeploy a storage space to a different target HBA using iSeries Navigator, follow these steps:

1. (Optional) Shut down the hosted server to which the storage space you want to redeploy is linked. It is not a requirement to shut down the server unless you want to redeploy the boot drives, or the drive you want to redeploy is being used by a Windows application.
2. Expand **Integrated Server Administration**.
3. Select **All virtual disks**.
4. To determine the current target HBA to which the storage space (virtual disk) is linked:
 - a. Right-click the virtual disk.
 - b. Select **Properties**.
 - c. Select the **Links** tab and you see a window similar to the one shown in Figure 9-65. In this example, the virtual disk drive KPAX0663 is linked to network server description KPAX066 using storage path NHLIN07.

We want to redeploy the virtual disk to storage path NHLIN06.

Click **Cancel**.

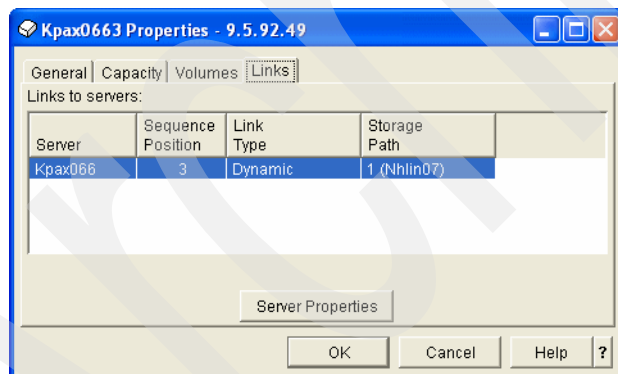


Figure 9-65 Determining the current target HBA (storage path)

5. To redeploy the virtual disk to a different target HBA, you first need to unlink it:
 - a. Right-click the virtual disk.
 - a. Select **Remove link**.
 - b. Click **Remove**.

The drive is unlinked from both the storage path and network server description to which it was previously linked (deployed).

6. To redeploy the virtual disk to a different target HBA, right-click it and select **Add link**. The panel in Figure 9-66 on page 439 appears.

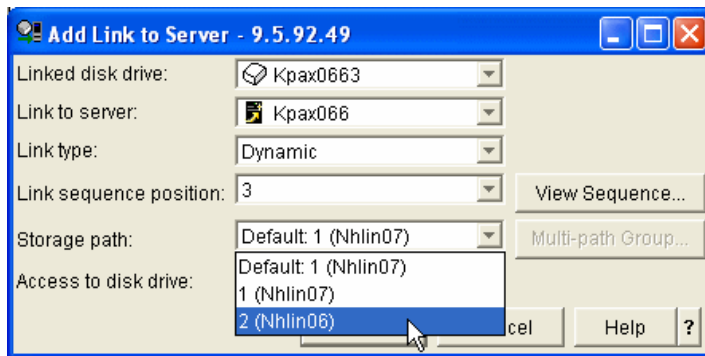


Figure 9-66 Redeploying a virtual disk to a different storage path (target HBA)

7. From the drop-down Link to server menu, select the network server description that you want to relink the storage space to. We are redeploying the storage space to a different storage path on the same NWSD, so select the same server that it was linked to before.
8. From the drop-down Storage path menu, select the storage path (target HBA) that you want to redeploy the virtual disk to, NHLIN06 in this case.
9. Click **OK**.

As previously mentioned, if you redeploy the boot drives from one target HBA to another, you must also clear persistent data on the initiator HBA port that is acting as the boot port. In this case, from either the server's local console, or from a Remote Control session on the Service Processor (or Management Module), follow these steps:

1. Power on the xSeries or Blade server that corresponds to the NWSD that you have just changed.
2. While the server is powering up, press CTRL-Q when you see the following line on the display:
Press <CTRL-Q for Fast!UTIL>
3. After a minute or two, you see a display similar to the one shown in Figure 9-67 on page 440.

Note that in this example, configuration was performed remotely using a Remote Control session on the Service Processor of an xSeries server.

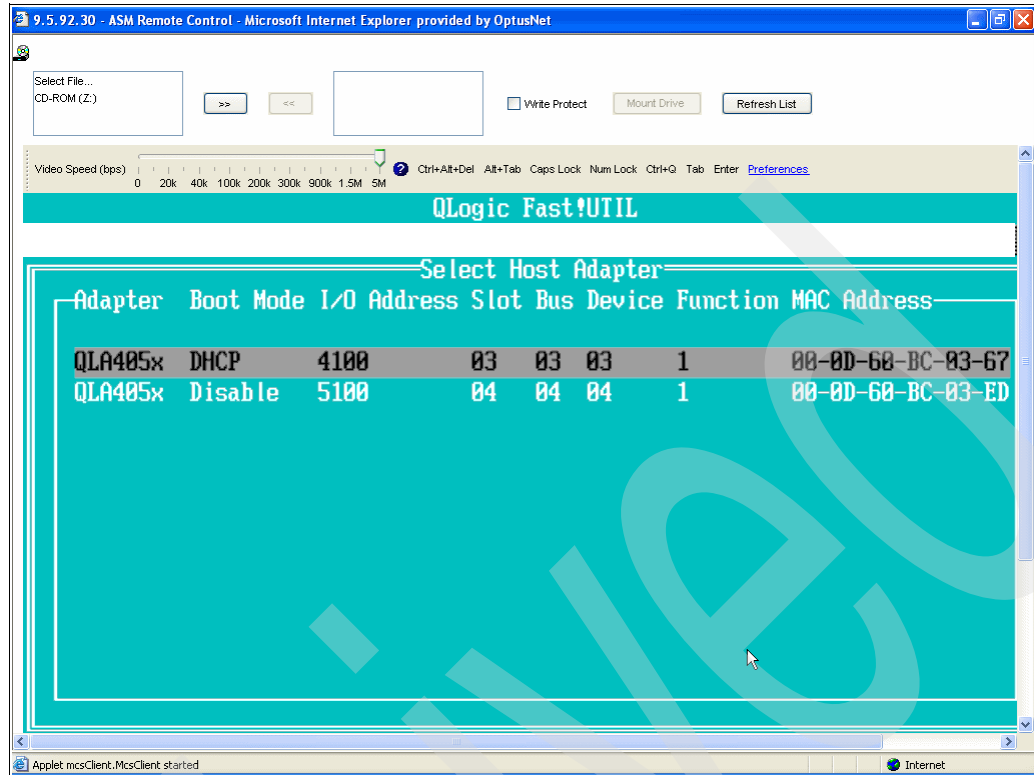


Figure 9-67 QLogic Fast!UTIL display

Figure 9-67 shows the initiator HBA ports that are installed in the xSeries or Blade server, their MAC addresses, and other configuration information. In Figure 9-67, there are two ports; the boot port is the one with Boot Mode set to DHCP.

4. Highlight the boot port by pressing the down arrow (if necessary) and then Enter.
5. The Fast!UTIL Options menu is displayed. From now on, any settings you change are changed only for the port you selected on the previous display.
Select **Configuration Settings**. Press Enter.
6. The Configuration Settings menu shown in Figure 9-68 on page 441 appears. You navigate your way around this utility by using the Up and Down arrow keys to select a menu option, then press Enter. To back up, press the Escape key.

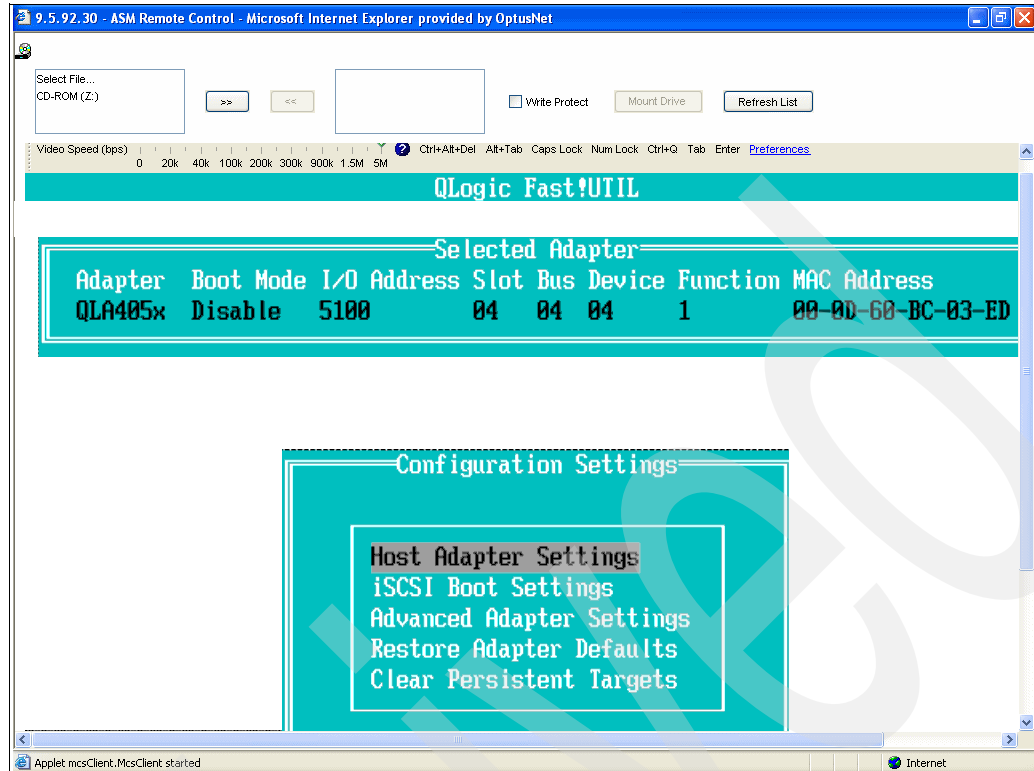


Figure 9-68 The Configuration Settings menu

7. From the Configuration Settings menu, select **Clear Persistent Targets**. Press Enter. Clear the Persistent Targets and then Escape back to the Configuration Settings menu.
8. Press Escape until you are prompted to **Save changes**. Press Enter. This takes a couple of minutes.
9. From the Fast!UTIL Options menu select **Exit Fast!UTIL**. Press Enter.
10. You are prompted to **Reboot System**. Press Enter. After the xSeries or Blade server has rebooted, power it off.
11. Assuming that you have already installed Windows on this server, start it up from iSeries Navigator.

The storage space has now been redeployed to the same network server description using a different target HBA. This is another way of saying that the storage space has now been redeployed to the same hosted server using a different storage path.

9.15.2 Redeploying a Virtual Ethernet LAN to a different target HBA

To redeploy a Virtual Ethernet LAN from one target HBA to another, you simply change the NWSH that the Virtual Ethernet LAN is assigned to. Note that when we redeploy a Virtual Ethernet LAN from one target HBA to another, we are not changing the hosted server (network server description) that the Virtual Ethernet LAN is assigned to, we are simply changing the data path that the Virtual Ethernet LAN uses to send data to the hosted server.

Using iSeries Navigator

To redeploy a Virtual Ethernet LAN to a different target HBA using iSeries Navigator, follow these steps:

1. Shut down the hosted server.
2. Expand **Integrated Server Administration**.
3. Select **Servers**.
4. Right-click the hosted server to which the Virtual Ethernet LAN that you want to redeploy is currently deployed. Select **Properties**.
5. Select the **Virtual Ethernet** tab as shown in Figure 9-69. You see all the Virtual Ethernet LANs deployed on this hosted server. In this example, Virtual Ethernet line KPAX066V1 is currently deployed to NWSH (target HBA) NHLIN06.

We want to redeploy the Virtual Ethernet LAN to NWSH NHLIN07.

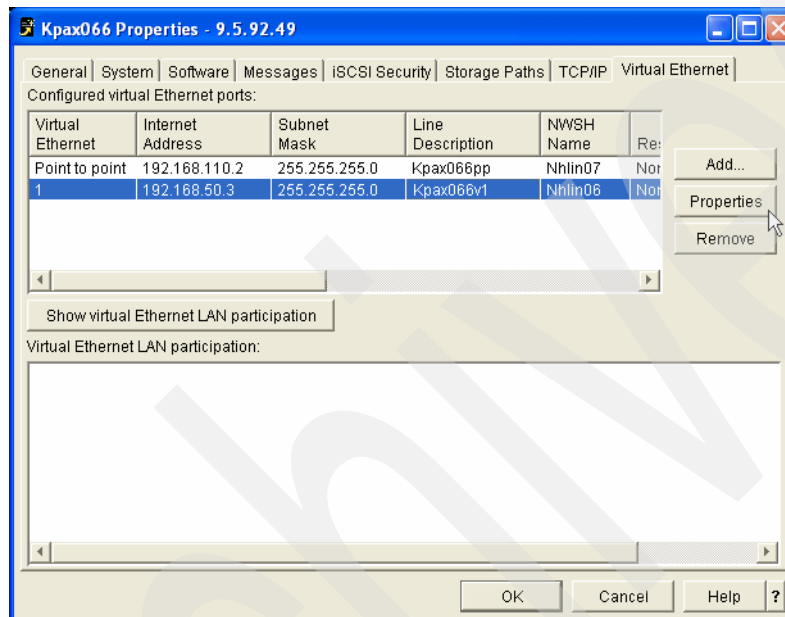


Figure 9-69 Redeploying a Virtual Ethernet LAN to a different target HBA - 1

6. Select the Virtual Ethernet LAN to be redeployed and click the **Properties** button. You see the panel shown in Figure 9-70.

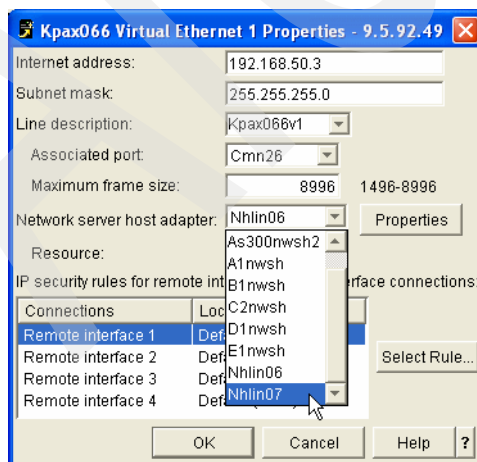


Figure 9-70 Redeploying a Virtual Ethernet LAN to a different target HBA - 2

7. From the drop-down Network server host adapter menu, select the new NWSH (target HBA) that you want to redeploy the Virtual Ethernet LAN to, NHLIN07 in this example. Click **OK**.
8. Click **OK** on the Properties window.

The Virtual Ethernet LAN has now been redeployed to the same network server description using a different target HBA.

9.15.3 Changing the boot port

If you have more than one initiator HBA port in your hosted server, you need to tell i5/OS Integrated Server Support which initiator HBA port Windows is booting off. (If there is only one initiator HBA port in the hosted server, you do not have to do this because there is no ambiguity.)

We have already covered how you can set up the boot port as one of the tasks you need to complete when *adding* an initiator HBA port to an xSeries or Blade server. However, you might also want to simply *change* (redeploy) the boot port from one initiator HBA port to another, having previously set up your initiator HBA ports. We therefore document the procedure to *change* the boot port here, as a stand-alone task.

Using iSeries Navigator

To change the boot port, you need to:

- ▶ Reconfigure both the new boot port and the old boot port using the CTRL-Q utility.
- ▶ Record the new boot device information.
- ▶ Specify the new boot device information in the remote system configuration object (RMTSYS) for the hosted server.

To change the boot port using iSeries Navigator, follow these steps:

1. Shut down the hosted server.
2. Configure the boot and non-boot ports using the CTRL-Q utility as described in 9.14.7, “Configuring the additional initiator HBA port as the boot port” on page 429 and then return here. Although this procedure is written to configure a *new* initiator HBA port as the boot port, it is exactly the same procedure if you want to *change* the boot port.

Record the following information for the (new) boot port:

- Bus:
- Device:
- Function:

3. You now need to change the boot port in the remote system configuration object.

Expand **Integrated Server Administration** → **iSCSI Connections** → **Remote Systems**.

4. Right-click the Remote system whose boot port you want to change and select **Properties**.
5. Select the **Boot Parameters** tab. You see a window similar to the one displayed in Figure 9-71 on page 444.

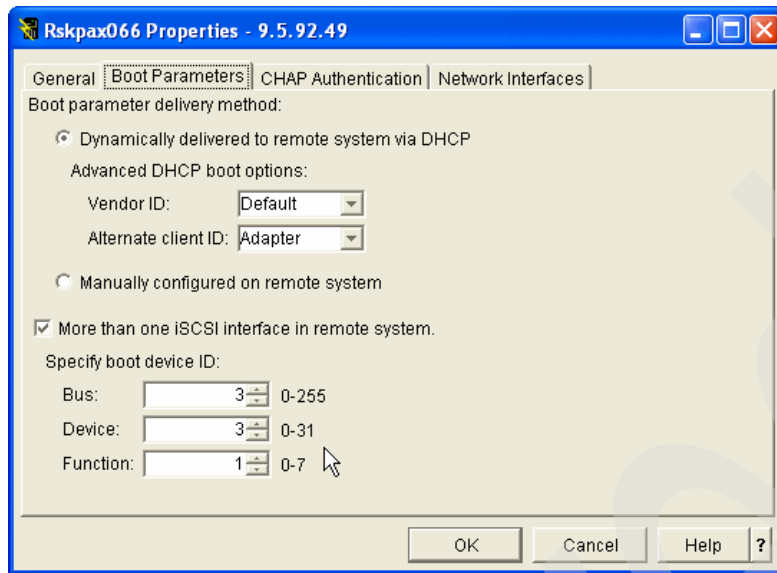


Figure 9-71 Changing the boot device ID (boot port) in the RMTSYS configuration object

6. Make sure that the **More than one iSCSI interface in remote system** check box is checked. (It should already be checked, because you are changing the existing setting.) Note that the term “iSCSI interface in remote system” is another term for an initiator HBA port.
7. Enter the values you determined for the new boot port using the CTRL-Q utility in the Bus, Device, and Function parameters.
8. Click **OK** to save the Properties.

Note: Specifying the boot device ID information in the RMTSYS configuration object is in fact optional because these values are overridden by the values set in the CTRL-Q utility. However, a future enhancement to System i™ Integrated Server Support will allow you to specify values for the boot device ID through the panel shown in Figure 9-71 and override the CTRL-Q settings. You will therefore not need to run the Ctrl-Q utility to specify the boot device ID.

We recommend that you enter the CTRL-Q values in the RMTSYS configuration object as described in this section for consistency and documentation purposes.

The hosted server should now boot from the new initiator HBA port that you specified as the boot port.

9.15.4 Preventing a non-boot drive from using an initiator HBA port

You can configure an iSCSI HBA port in a hosted server with a SCSI IP address, a LAN IP address, or both types of IP addresses. The presence of a SCSI IP address enables SCSI (disk) traffic, and the presence of a LAN IP address enables Virtual Ethernet LAN traffic.

You can prevent a non-boot virtual disk from using a particular initiator HBA port by disabling that port for SCSI connections. You do this by blanking out the SCSI IP address information for the port in the Network interface of the RMTSYS object.

While you can specify which initiator HBA ports you do *not* want virtual disks to use, you cannot deploy a particular virtual disk to a specific initiator HBA port, except for the boot port.

Important: When using this technique to deploy virtual disks, be careful not to disable the boot port. There is no way of telling which port is the boot port through iSeries Navigator. You must use the CTRL-Q utility to determine the boot port in a hosted server with multiple initiator HBA ports.

This technique can also be used in the following situations:

- ▶ To limit those initiator HBA ports, which Virtual Ethernet LANs can use. However, there is a better method to do this, which we describe in 9.15.5, “Redeploying a Virtual Ethernet LAN to a different initiator HBA port” on page 446.
- ▶ To limit those target HBAs, which storage spaces and Virtual Ethernet LANs can use. This is not a good idea because it would greatly reduce your flexibility in deploying storage spaces and Virtual Ethernet LANs on the target side. A much better way to deploy storage spaces and Virtual Ethernet LANs on the target side is described in 9.15.1, “Redeploying a storage space to a different target HBA” on page 437 and 9.15.2, “Redeploying a Virtual Ethernet LAN to a different target HBA” on page 441.

Using iSeries Navigator

To prevent a non-boot drive from using a particular initiator HBA port, follow these steps:

1. Shut down the hosted server.
2. Expand **Integrated Server Administration** → **iSCSI Connections** → **Remote Systems**.
3. Right-click the remote system (hosted server) whose initiator HBA ports you want to restrict. Select **Properties**.
4. Select the **Network Interfaces** tab as shown in Figure 9-72.

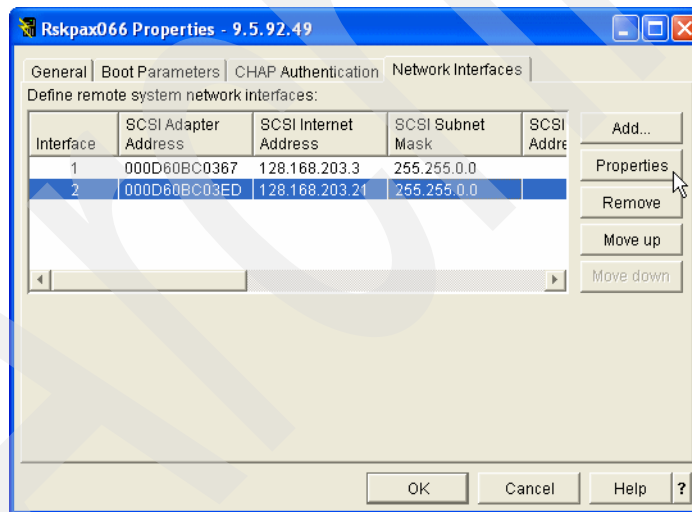


Figure 9-72 Disabling the SCSI side of an initiator HBA port - 1

5. Select the interface (initiator HBA port) that you want to disable and click **Properties**.
6. Blank out the Remote SCSI interface IP addressing information as shown in Figure 9-73 on page 446. Be careful not to disable the boot port!

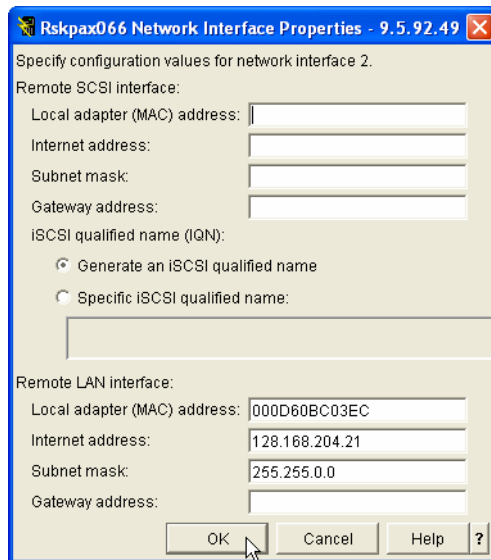


Figure 9-73 Disabling the SCSI side of an initiator HBA port - 2

7. Click **OK**. The interface is saved as shown in Figure 9-74. Notice that the SCSI IP addressing information for Network interface 2 is missing.

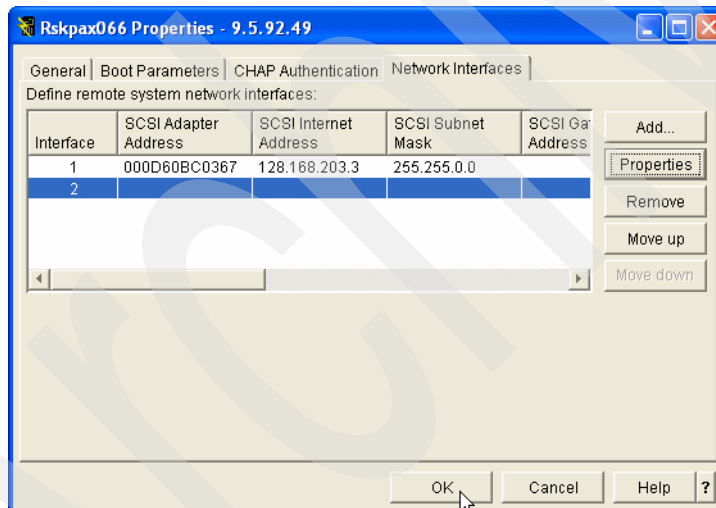


Figure 9-74 Disabling the SCSI side of an initiator HBA port - 3

8. Click **OK** to save the Properties.
9. Start the hosted server.

You have now prevented virtual disks from using one of the initiator HBA ports on the hosted server.

9.15.5 Redeploying a Virtual Ethernet LAN to a different initiator HBA port

In contrast to virtual disks, you *can* redeploy a Virtual Ethernet LAN to a specific initiator HBA port by changing the properties of the relevant Virtual Ethernet connection in Windows.

First, we browse the Windows Network Connections to gain an understanding of the relationship between the iSCSI network interfaces and the Virtual Ethernet LANs that are configured on this hosted Windows server. Then we work through an example of how you can specify which initiator HBA port the Virtual Ethernet LAN should use.

Browsing Windows Network Connections

We provide an example of an xSeries server with two initiator HBA ports installed.

To browse Windows Network Connections, follow these steps:

1. On the Windows server console, click **Start** → **Control Panel**.
2. Right-click **Network Connections**. Select **Open**. You see a display similar to the one shown in Figure 9-75.

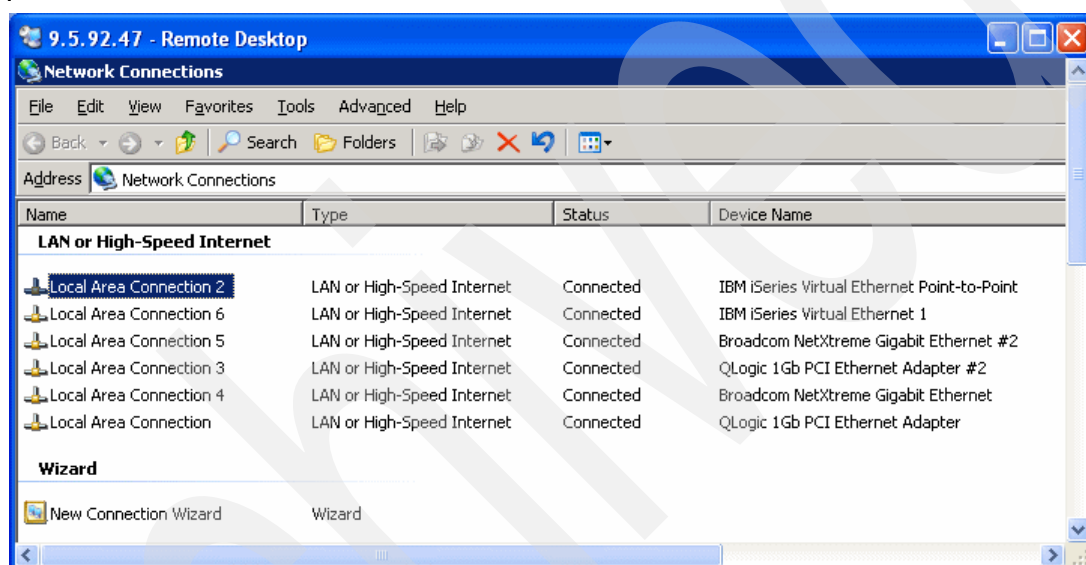


Figure 9-75 Browsing Windows Network Connections (xSeries) - 1

In the example shown in Figure 9-75, you see the following connections:

- Two QLogic 1 Gb PCI Ethernet Adapters (connections 1 and 3)
These correspond to the two iSCSI initiator HBA ports installed in the hosted server.
- Two BroadCom NetXtreme Gigabit Ethernet adapters (connections 4 and 5)
These correspond to the two external Ethernet LAN ports in the xSeries server.
- One point-to-point Virtual Ethernet LAN (connection 2)
- One (non point-to-point) Virtual Ethernet LAN (connection 6)

Note that the Windows end of the Virtual Ethernet LAN connections (2 and 6) are shown using separate icons, even though they are actually deployed on the two QLogic 1 Gb PCI Ethernet Adapters (connections 1 and 3). By default, the two Virtual Ethernet LAN connections are automatically deployed to the two initiator HBA ports (connections 1 and 3) using autodeployment. There is no way to tell which Virtual Ethernet LAN is deployed to which initiator HBA port. However, you can force a Virtual Ethernet LAN to use a specific initiator HBA port, which we describe below.

Redeploying a Virtual Ethernet LAN to a different initiator HBA port

To switch a Virtual Ethernet LAN from one initiator HBA port to another, follow these steps:

1. On the Windows server console, click **Start** → **Control Panel**.
2. Right-click **Network Connections**. Select **Open**. You see a display similar to the one shown in Figure 9-76.

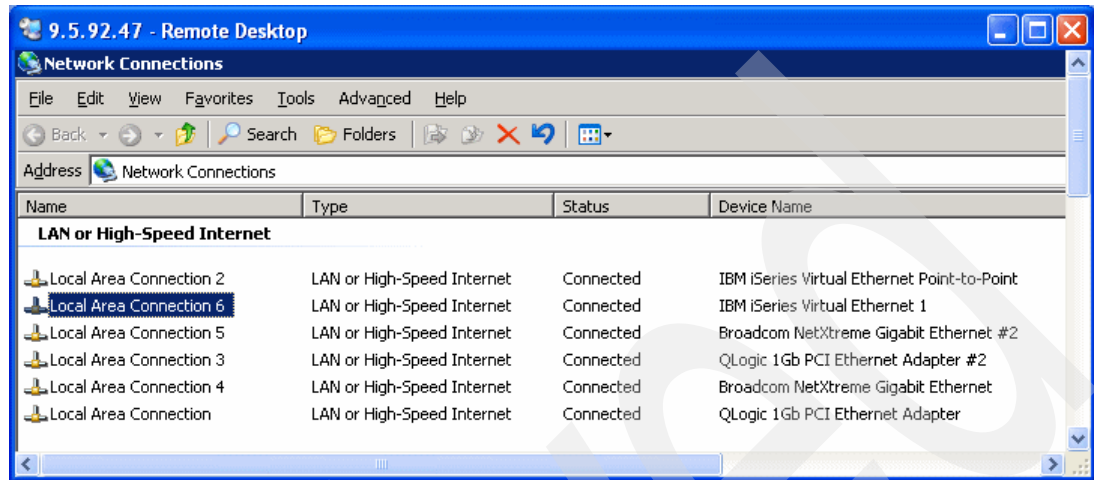


Figure 9-76 Viewing the Windows network connections

3. Explore the QLogic 1 Gb PCI Ethernet Adapter that you want to deploy the Virtual Ethernet LAN to as follows:
 - a. Double-click the QLogic adapter, then click **Properties**.
 - b. Double-click **Internet Protocol (TCP/IP)** and you see a window similar to the one shown in Figure 9-77.

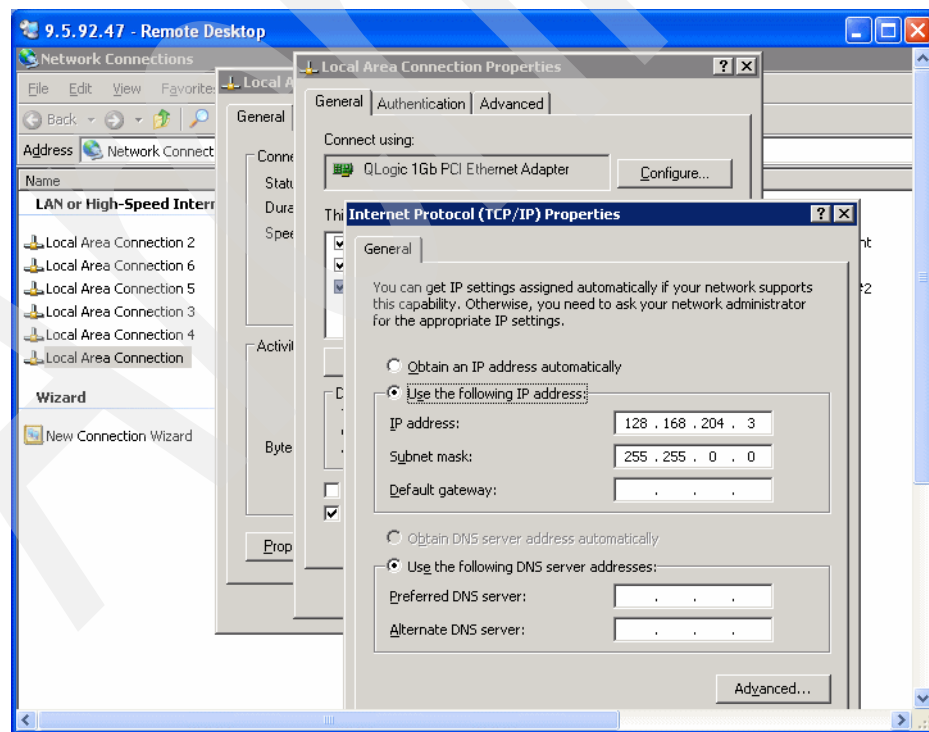


Figure 9-77 Browsing Windows Network Connections (xSeries) - 1

Figure 9-77 on page 448 shows the IP address of the Virtual Ethernet LAN side of the initiator HBA port. This is the address you assigned to it when you configured the network interface in the RMTSYS configuration object. Take note of the IP address, because you need it later in order to redeploy the Virtual Ethernet LAN to this initiator HBA port.

- c. Close the windows and return to the Windows Network Connections display.
4. (Optional) We explore one of the Virtual Ethernet LAN connections shown in Figure 9-76 on page 448 as follows:
 - a. Double-click either of the Virtual Ethernet LANs, then click **Properties**.
 - b. Double-click **Internet Protocol (TCP/IP)** and you see a window similar to the one shown in Figure 9-78.

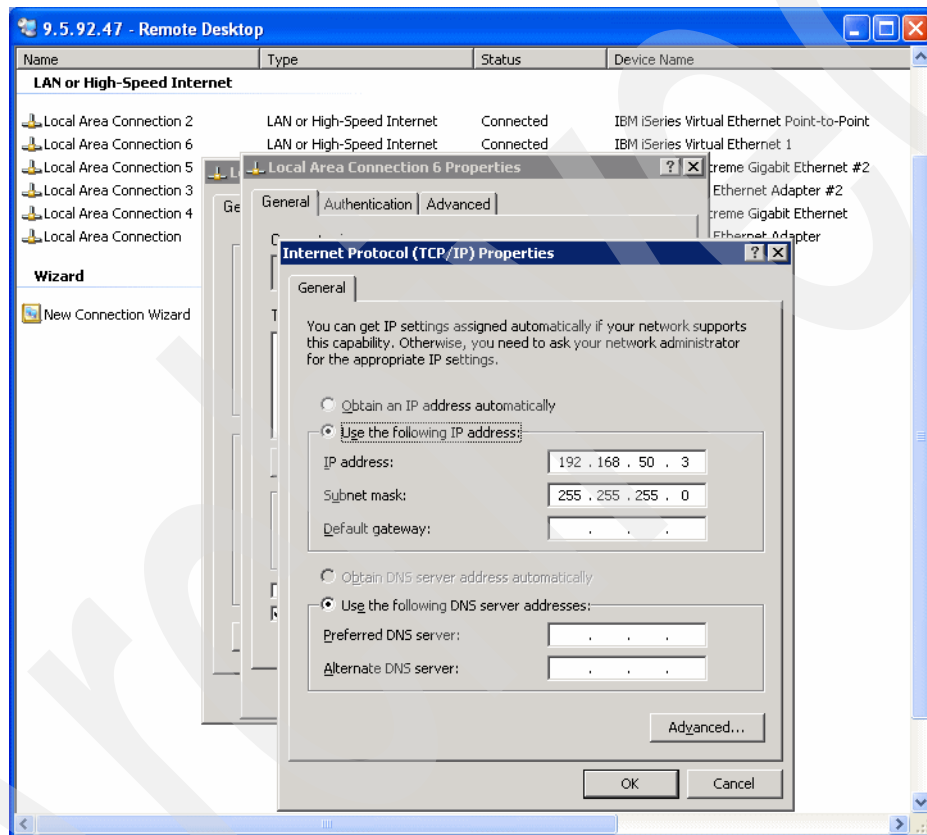


Figure 9-78 Browsing Windows Network Connections (xSeries) - 2

Figure 9-78 shows the IP address of the Windows end of the Virtual Ethernet LAN. This is the address that was automatically generated in the case of the point-to-point Virtual Ethernet LAN, or the address we assigned to it in the case of a non-point-to-point Virtual Ethernet LAN.

- c. Close the windows and return to the Windows Network Connections display.
5. Double-click the Virtual Ethernet LAN connection that you want to redeploy.
6. Click **Properties**. You see a display similar to the one shown in Figure 9-79 on page 450.

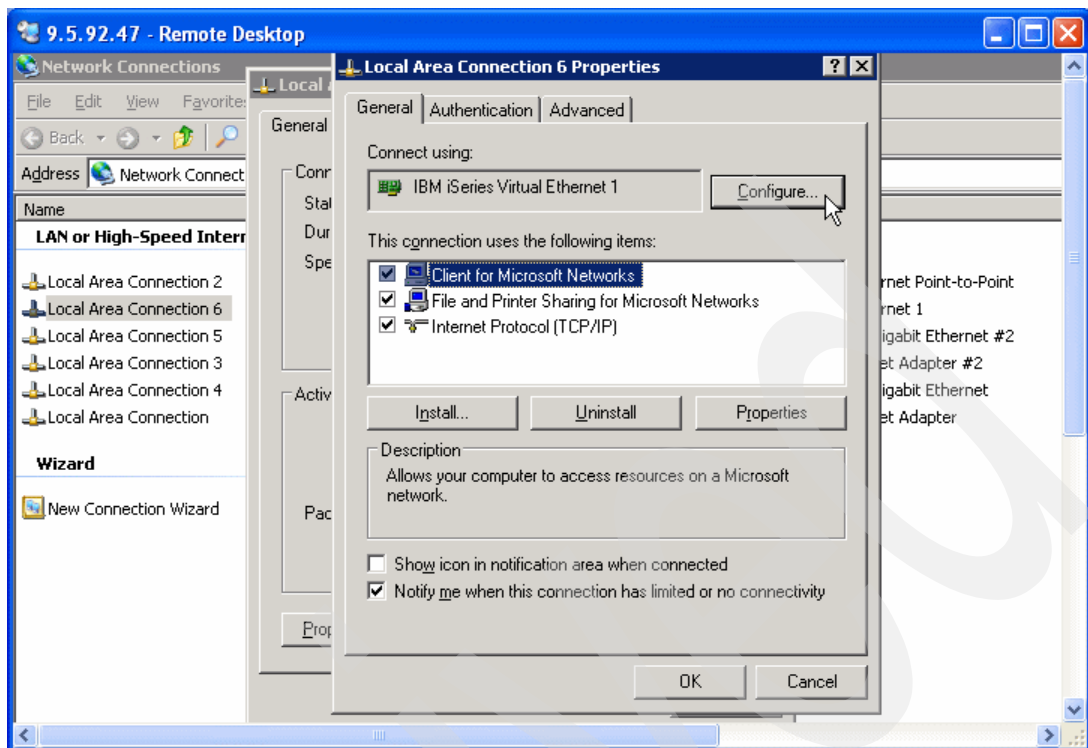


Figure 9-79 Redeploying a Virtual Ethernet LAN to a different initiator HBA port - 1

7. Click **Configure**.
8. Click the **Advanced** tab.
9. Click **Initiator LAN IP Address**.
10. Click the radio button next to the blank parameter and fill in the IP address of one of the QLogic network connections that you wrote down previously. An example is shown in Figure 9-80 on page 451.

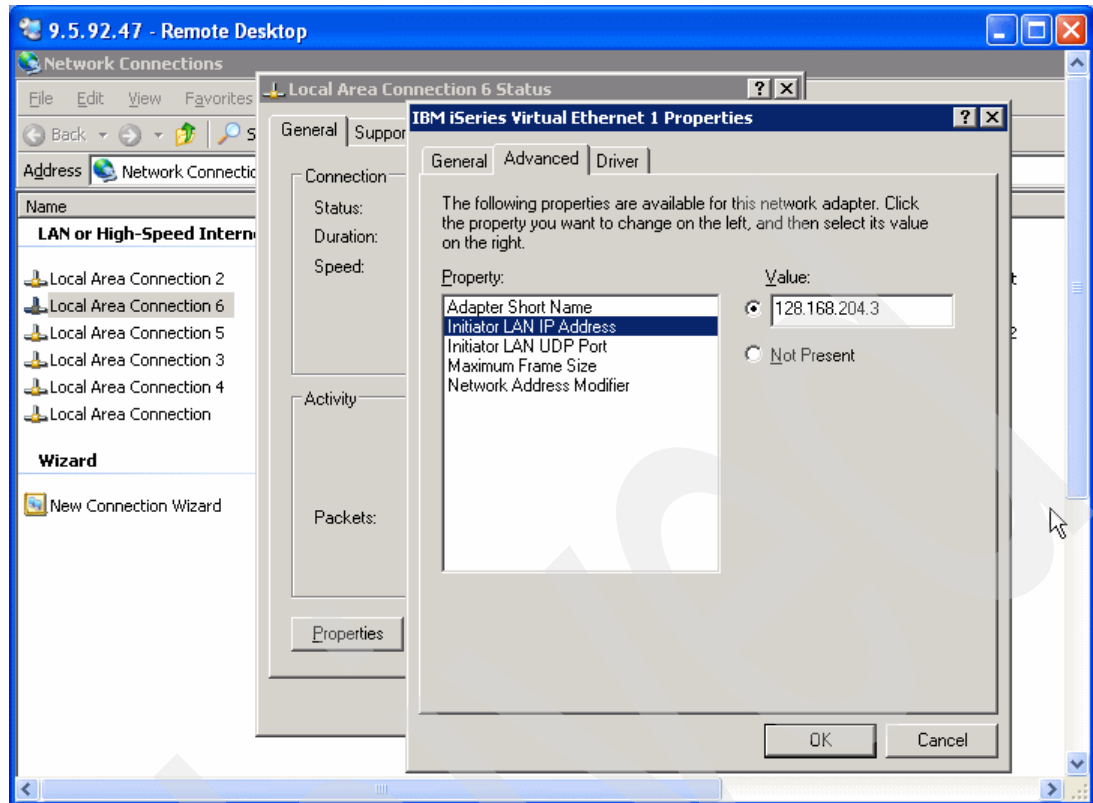


Figure 9-80 Redeploying a Virtual Ethernet LAN to a different initiator HBA port - 2

11. Click **OK**, then click **Close** to save your changes.

You have now deployed the Virtual Ethernet LAN to a specific initiator HBA port.

Archived

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 and Redpapers

For information about ordering these publications, see “How to get IBM Redbooks” on page 454. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *IBM eServer i5 and iSeries System Handbook: IBM i5/OS Version 5 Release 3 October 2004*, GA19-5486
<http://www.redbooks.ibm.com/redpieces/abstracts/ga195486.html?Open>
- ▶ *IBM eServer i5, iSeries, and AS/400e System Builder IBM i5/OS Version 5 Release 3 - October 2005*, SG24-2155
<http://www.redbooks.ibm.com/redpieces/abstracts/sg242155.html?Open>
- ▶ *Logical Partitions on System i5: A Guide to Planning and Configuring LPAR with HMC on System i*, SG24-8000
<http://www.redbooks.ibm.com/abstracts/sg248000.html?Open>
- ▶ *Implementing IBM Director 5.10*, SG24-6188
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246188.pdf>
- ▶ *PCI and PCI-X Placement Rules for IBM System i5, eServer i5, and iSeries servers with i5/OS V5R4 and V5R3*, REDP-4011
<http://www.redbooks.ibm.com/abstracts/redp4011.html?Open>
- ▶ *Managing OS/400 with Operations Navigator V5R1 Volume 1: Overview and More*, SG24-6226
<http://www.redbooks.ibm.com/redbooks/pdfs/sg246226.pdf>

Online resources

These Web sites and URLs are also relevant as further information sources:

- ▶ eServer Hardware Information Center
http://publib.boulder.ibm.com/infocenter/iseries/v1r2s/en_US/index.htm
- ▶ iSeries Information Center
<http://publib.boulder.ibm.com/iseries/>
- ▶ For general integrated server information including supported models, supported versions of microsoft OS and tested service packs, latest fixes for the integrated servers, and much more, go to:
<http://www.ibm.com/servers/eserver/iseries/integratedxseries>
- ▶ Windows environment on iSeries (part of iSeries infocenter) at:
- ▶ <http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzahq/rzahq.pdf>

- ▶ IBM Director Installation and Configuration Guide at:
http://publib.boulder.ibm.com/infocenter/eserver/v1r2/topic/dirinfo/fqp0_bk_install_gde.pdf
- ▶ iSCSI Network Planning Guide
http://www-03.ibm.com/systems/i/bladecenter/pdf/iscsi_planning.pdf
- ▶ System p5 and i5 eServer p5 and i5 and OpenPower PCI adapters at:
<http://publib.boulder.ibm.com/infocenter/eserver/v1r3s/topic/iphak/iphak.pdf>
- ▶ System p5 and i5 eServer p5 and i5 iSCSI Host Bus Adapter
<http://www-03.ibm.com/systems/i/bladecenter/iscsi/>
- ▶ Ethernet switches for iSCSi
<http://www-03.ibm.com/systems/i/bladecenter/iscsi/switches.html>
- ▶ For xSeries and BladeCenter, see Configuration and Options Guide, SCOD-3ZVQ5W
ftp://frp.software.ibm.com/pc/pccbbs/pc_servers_pdf/cog.pdf
- ▶ For other BladeCenter specific information, see:
<http://www-03.ibm.com/systems/bladecenter/>

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

A

Auxiliary Storage Pool
 physical disk drive 260
Auxiliary storage pool (ASP) 30, 38, 152, 247, 260, 270, 274–275

B

Backup Domain Controller (BDC) 301
backup from a server-centric perspective 257
Backup Recovery
 and Media Services 257, 269–270, 272, 283
Backup Recovery and Media Services (BRMS) 257, 269–270, 272, 283
backup technique positioning and recommendations 261
Base Management Controller (BMC) 7
Baseboard Management Controller (BMC) 14, 29
basic iSCSI configuration 17, 19–20, 325, 343–344, 349, 352–353, 358–359, 372, 375, 401, 418, 437
Blade Center 1, 4, 6–7, 44–45, 49–51, 53–55, 65, 82, 92, 140, 143, 149–153, 155, 169–177, 181, 183, 185, 191–194, 196, 200–201, 211, 221, 229–230, 232–233, 236–237, 240–241, 454
 management module 6
 management processor 7
 Planning 49
 second port 45
Blade center
 correct Blade 193, 201
 Management Module 169
Blade server
 autodeployment works 375
 automatic deployment 385
 bandwidth requirements 340, 354
 initiator HBA ports 21
 iSCSI connection 325
 manual deployment 375
 manual deployment works 385
 second initiator HBA port 357, 418
 too little bandwidth 353
 too much bandwidth 353
 virtual Ethernet LAN connection 355
blade server 5–7, 12, 14, 16–21, 37, 66, 74, 91, 101, 151, 218, 325–327, 330–331, 333, 335–342, 352–362, 368, 370, 374–376, 378–379, 384–386, 388–389, 393, 400, 412, 418, 425–427, 429–430, 433, 436–437, 439–440, 443
 new disk 101
 Remote System object 66
BladeCenter 11–14, 17, 20, 23–24, 27–28, 34, 37, 74, 327–329, 331, 335–336, 338, 342, 346, 356–357, 359–362, 384, 387, 409, 425, 436
boot diskette 292
Boot drive 28, 328–329, 334, 339, 363, 365, 370–371, 373–378, 380–381, 383–384, 386, 436–439, 444

boot port 16, 20, 27–28, 347–348, 356, 358, 361–362, 365, 371, 373–374, 376, 380–381, 383–384, 386, 418, 426–431, 433–434, 436–440, 443–445

C

Challenge Handshake Authentication Protocol (CHAP) 6, 31–32
Change NWSH 254
CHAP Name
 CHAP Name 187, 194, 196
CHAP name 54, 69, 79, 93, 187, 193–194, 196–197, 199, 201–202
CHAP secret 32, 54, 69, 93–94, 187, 193–194, 196–197, 199, 201–202
CL Command 25, 38, 149, 156, 163, 165–169, 175–176, 182–184, 195–196, 198–201, 205–207, 220, 228, 232, 235, 237–241, 246, 251, 268, 294–295
CL Program 218, 220, 236, 261, 270, 275–276, 282, 286
CNNSEC 20–21, 23, 151, 156, 202–207, 225
command CRTDEVNWSH 164
 First page 164
 Second page 164
command line 54, 84, 110, 113, 135–137, 146–149, 154, 156, 165, 176, 200, 220, 228, 231, 235, 251
 command STRQSH 146–148
 following CL Commands 235
 following command 137
 type CFGTCP 220
 type CHGIPLA 220
 type CHGNWSCFG 184, 199, 206
 type CRTDEVNWSH 163
 type CRTNWSCFG 182, 196, 205
 type DLTNWSCFG 184, 207
 type DSPDEVD 165
 type DSPJOBLOG 200
 type DSPNWSCFG 183, 198, 206
 type INZNWSCFG 175–176
 type WRKCFGSTS 166–167, 228, 232, 235, 239
 type WRKDEVD 165, 168, 220
 type WRKNWSCFG 183, 198, 200–201, 206
 type WRKNWSD 228, 235
 Type WRKNWSSTG 251
 type WRKNWSSTG 246
 type WRKNWSSTS 239
command, CL
 SAVLICPGM (Save Licensed Program) 270
commercial processing workload (CPW) 112
complete disaster recovery backup 262
complete file level backup 262
Ctrl-Q utility 186–187, 193, 196, 201, 365, 426, 433–434, 443–445

D

device description 6, 57, 151–152, 156, 163–165, 169,

- 220, 254, 267, 281, 295
- disaster recovery backup 262
- disaster recovery backup methods
 - copy to disk 272, 283
 - save to disk 272, 283
 - save to tape 283
- disaster recovery restore special cases
 - recreating TCP/IP interfaces 281
 - relinking user storage spaces 281
 - restoring configuration objects 281
- disk drive 24, 28–30, 38, 45, 149, 151, 241–246, 248, 251–254, 256–257, 260–262, 365–367, 390, 392, 399
 - current Default path 366
 - Default path 216
- disk space 13, 112, 268
 - large amount 268
- Diskpart utility 244–246, 248–251
- Dynamic Host Configuration Protocol (DHCP) 35–36

E

- Ethernet network 12, 14, 17–18, 20, 24, 28–30, 33, 40, 150, 152–153
 - i5 system 150
 - i5/OS communicates 152
 - management module 28
 - other systems 17

F

- file level 258, 260, 290–291
- file level backup 262
- File Transfer Protocol (FTP) 300, 303
- firewall 114

G

- Gateway address 85, 89, 94, 158, 163–164, 188, 197–198, 200, 404, 423, 435

H

- Hardware Management Console (HMC) 307, 309, 311–312, 315
- hardware managment console (HMC) 14, 89
- Hierarchical Storage Manager (HSM) 270
- high speed link (HSL) 9–11, 17, 23
- Host adapter
 - storage path 56–57
 - VE path 56–58
- Host Bus Adapter 13
- host bus adapter (HBA) 5, 13–27, 32, 34–38, 40, 57, 219–220, 325–327, 329–337, 339–367, 369–370, 372–375, 377–386, 388–391, 393, 395–397, 399–419, 421–432, 434–439, 441–442, 444–447, 449–451
- Host Bus Adpater (HBA) 13–14, 16–18, 21, 23–24, 26–27, 32, 34, 36–37, 40
- hosted Windows server
 - adequate bandwidth 331
- hosting i5/OS partition
 - storage space 329, 367
- hot spare system drive 280

I

- i5/OS 9, 14, 16–21, 23–24, 26–34, 36–38, 40, 149–153, 207, 210, 212–213, 216, 225, 241, 255–257, 259–263, 265–267, 269–272, 275–276, 278, 280, 282, 286–287, 289–303, 306–312, 314–315, 321–323, 325, 327–332, 334, 336–337, 339, 342–344, 346, 348, 351–355, 357, 363, 365–367, 389, 399, 402–403, 407–408, 410–414, 416, 426, 429, 433, 436, 443
- i5/OS partition 17–18, 20–21, 37, 261, 266, 269, 291, 295, 297, 300, 309–313, 325, 328–331, 335, 342–343, 353, 367, 403, 408, 412–414
 - basic iSCSI connection 17
 - Ethernet adapter 328
 - tape drive 295
 - tape drives 295
- i5/OS-centric backup 257, 260, 265–266, 270, 272, 275
- i5/OS-centric recovery 278, 282
- IBM Director
 - following versions 114
- IBM Director (ID) 14, 20, 28–30, 47, 49–50, 53–55, 65, 81–82, 107–116, 118–119, 122, 125–126, 128–129, 132, 134–136, 138–140, 146–148, 454
- ID and password (IP) 12–14, 27–28, 30, 32–34, 36, 39–40, 109–111, 127, 139, 141, 143, 306–308, 310, 314, 322–323
- incremental or differential file level backups 262
- Independent Asp 86, 152, 272, 274, 284–285
- independent Asp
 - device name 272, 274, 284–285
- Initiator HBA 7, 13–14, 16–17, 21, 37, 325, 327, 330–341, 344–349, 351–358, 360–365, 367–377, 379–386, 388–391, 393, 395–396, 399–405, 407–415, 418, 425–433, 436–437, 439–440, 443–447, 449, 451
 - first port 16
 - port 16, 327, 330–331, 336–337, 347–348, 356–357, 362, 370, 382, 385, 401, 409–410, 418, 426, 432, 436, 443, 445, 447
 - second port 16
- initiator HBA
 - boot port 365
 - Port 1 374
 - port layout 335
 - second port 335, 356–357, 375
- Initiator HBA port 331, 336, 347–348, 356–357, 362, 382, 385, 409–410, 418, 426, 443
- initiator HBAs
 - maximum number 334
- initiator port 7, 21, 347, 356, 361, 406, 429
 - second port 356, 361, 429
 - virtual Ethernet LAN 347
- Installation drive 152, 263, 267–268, 276–277, 281, 287–288
- INSWNTSVR command 3, 29, 53, 55, 86, 153, 218, 309
 - Screen 4 86
- Integrated File System (IFS) 2, 6, 30, 85, 136, 138, 145, 301
- integrated files system (IFS) 241
- Integrated Server 1–7, 23–25, 27, 29–30, 34–35, 37–38, 43–44, 47–48, 50–51, 53–55, 58, 61, 65, 71, 75, 78, 80, 86, 94, 99, 149, 151–156, 158–162, 172–174, 177–181,

- 185, 188, 190–192, 196–197, 200, 202–204, 207–208, 210, 212–213, 215–222, 224, 226–229, 231–239, 241–242, 244–246, 248, 250–254, 264–265, 306, 308–312, 315, 321–323, 329, 403, 420, 424, 433–434, 436, 438, 442–443, 445, 453
 - virtual ethernet 3
- integrated server
 - catastrophic failure 265
 - line description 321
 - memory intensive work 37
 - remote system configuration object 27
 - Right click 227, 233
 - storage capacity 152
 - TCP/IP domain name 216
 - TCP/IP hostname 216
 - type NWSD name 250
 - unique features 54
 - web site 44
- Integrated Windows server
 - View Ethernet connections 323
- integrated Windows server
 - complete image 266
 - important component 266
 - individual components 272, 283
 - System i5 tape drive 293
 - unattended storage space backup 261
- integrated windows server 27, 149, 152, 231, 256–257, 260–261, 266–267, 269, 272, 276, 279–283, 286, 291–293, 295, 297, 299–300, 306–308
- Integrated xSeries Adapter (IXA) 9, 34, 150, 155, 256, 308
- Integrated xSeries Server
 - Later models 279
- Integrated xSeries server (IXS) 2, 9, 34, 150, 155, 256, 258, 261, 264–268, 278–280, 290, 292, 294, 300–301, 308
- Internet address 24, 27, 78, 85–86, 89, 93–94, 158, 163–164, 177, 179, 183–184, 188, 192, 196–198, 200, 217, 323, 423
 - Internet address 158, 163–164, 196–197
- inter-partition connection 309–312
- INZNWSCFG described (ID) 169–179, 184, 186–187, 194, 196–199, 202, 212, 222, 229–230, 247
- ip address (IA) 7, 27, 32–36, 39–40, 66, 115, 141, 143, 153, 282, 302, 306, 308, 310, 314, 321–323, 372–373, 375–376, 383, 386
- IP Security
 - Rule 79, 87–88, 92, 205–207, 214, 216–217, 254
 - Rules tab 203
- iSCSI 1–7, 9–21, 23–25, 27–36, 38–40, 43–45, 47–51, 53–59, 61, 64–66, 71, 75, 77–80, 85, 87–89, 91, 93–94, 99, 104, 149–156, 158–162, 172–174, 177–181, 185–188, 190–194, 196–198, 200–204, 207–210, 214–216, 218–221, 225–226, 228–229, 231–233, 235–241, 247, 252–254, 306–307, 309, 315, 322, 325–331, 334–335, 339–344, 346, 348–350, 352–355, 357–359, 362–364, 367, 370, 372, 374–377, 379, 382, 385, 387–388, 390–399, 401–403, 405–406, 411, 413, 415–419, 425–430, 432, 434–437, 444, 447
- iSCSI adapter 14, 27, 33–34, 39
 - IP address 34
- iSCSI architecture 9–11, 17, 59, 325, 330
 - big difference 9
- iSCSI concept 152, 232
- iSCSI connection 3, 7, 12–14, 17, 19, 24, 46, 140, 154, 156, 162, 185, 192, 202–203, 218–219, 225, 331, 343, 352–353
 - iSeries side 24
 - vary-on process 219
- iSCSI environment 2, 107–108, 225, 256
 - IBM Director Server 107–108
- iSCSI HBA 13–14, 24–27, 32, 34, 36, 40, 49, 151, 236, 315, 322, 327, 340, 363, 391, 410, 436
 - direct connection 14
- iSCSI HBAs 23–24, 27, 32, 36, 38, 151
 - physical network 32
- iSCSI integrated server 2–3, 43, 51, 107–108, 140, 149, 152–153, 155, 221–222, 231, 236, 254
 - reference new terms 6
 - Status 221, 236
- iScsi integrated server
 - specific documentation 44
- iSCSI network 7, 9, 12–15, 17–22, 27, 29–31, 33, 36–37, 39–40, 44, 50–51, 54, 64–65, 151, 305–307, 314, 325–332, 335, 337, 340, 343–345, 352–354, 363–365, 369, 372, 374–375, 377, 379, 387–391, 393–394, 396–397, 399–401, 411, 413, 415, 417, 419, 425, 429, 433, 435, 437, 447, 454
 - Blade server 17
 - different components 332, 337
 - excess bandwidth 344, 354
 - i5/OS partition 20
 - initiator HBAs 15, 18
 - IPsec security 21
 - Other components 29
 - point-to-point virtual Ethernet LAN 374, 377
 - System i5 9, 12
 - VE LAN data 14
 - virtual Ethernet 39–40
 - virtual Ethernet data flows 151
 - virtual Ethernet LANX 374
 - virtual Ethernet LANY 377
- iSCSI part 188, 196–197
 - Adapter Address 196–197
- iSCSI qualified name (IQN) 6, 329, 406, 411–412, 416, 424, 435
- iSCSI Security
 - tab information 214
- iSeries 2–11, 13–14, 23–24, 27–28, 31–32, 34–38, 40, 44–45, 47, 49, 51, 107, 109, 112–115, 139, 141, 149, 151, 153–156, 169, 210–212, 219–220, 231–232, 254, 307, 309–310, 312–313, 315, 322
- iSeries Navigator 2–3, 25, 35, 40, 47, 58, 60, 65, 71, 77–78, 91, 99, 108–109, 111–112, 114, 136, 138, 149, 154–156, 158–162, 166, 169, 172–174, 177–180, 185, 190–193, 195, 202–205, 207–208, 219, 221, 224–226, 231–233, 237–241, 246, 251–254, 310–312, 321, 419, 424, 433, 438, 441, 443, 445
 - Basic Operations 114
 - boot port 445

- Detail Status integrated server 239
- integrated server administration 155, 225
- iSCSI integrated server 224
- New Connection Security Configuration 203
- new storage spaces 99
- Properties CNNSEC 204
- iSeries server 35–36, 38, 141, 149, 315
 - usually multiple disks 38
- IXA card 23, 151

K

- Keyboard, Video and Mouse (KVM) 193, 229
- keyboard/video/mouse (KVM) 28

L

- Last Known Good configuration 292
- licensed program product (LPP) 2, 54, 269, 272, 283
- licensed program saving 270
- line description 57, 77, 86, 152, 217, 219, 266, 271, 273–274, 281–282, 284–285, 308–309, 311–312, 314, 322–323
 - different types 271, 273–274, 284–285
 - i5/OS side 309

M

- mac address 7, 34, 36, 54, 403, 408–414, 427, 430, 440
- Management Module
 - Remote Control 229
 - Remote Control session 436
- management module 6–7, 14, 20, 24, 28, 54, 65, 74, 79, 143, 149, 151–152, 169, 181–182, 221–222, 229, 236, 241, 327–329, 426, 429, 436, 439
- Maximum transmission unit (MTU) 39–40, 314
- message queue 63, 86–87, 89, 91, 157, 159, 163–164, 166, 213–214, 238
- Multi-path group 24, 30, 56, 216
- must also configure (MAC) 34, 36, 39

N

- Network File System (NFS) 300, 302–303
- Network server
 - configuration 6, 27–28, 35, 55–56, 65, 169, 175–177, 182–185, 196–197, 199, 205–207
 - host adapter object 156, 158, 163
 - storage space link 248
 - type 55
- Network Server Description (NWSD) 6, 19–21, 23–24, 26–27, 29–30, 34–36, 256, 259–261, 263–267, 271–278, 281–288, 293
- Network server host
 - adapter object 156, 158, 163, 165
- Network server host adapter
 - object 151, 161, 215, 217
- Non-boot drive 328–329, 371–373, 375–377, 383, 386, 437, 444–445
- NTAP directories 268
- NWS Configuration 74, 91, 156, 175–177, 182–185, 196–201, 205–207, 232

- nwscfg object 59, 77, 225
- NWSD 267
- NWSD type 209
- NWSH object 23–25, 36, 64, 151, 156, 159–160, 163, 220, 327, 333, 338, 342, 346, 355, 365–366, 371–372, 374–375, 377, 403, 405
- NWSHs 21, 24
 - different virtual Ethernet ports 24

O

- operating system
 - incorporated functions 2
 - Integrated Server Support 54
- operating system (OS) 2–3, 16, 28, 47, 54–55, 58, 86–87, 89–90, 94, 152, 191, 209, 212, 236–237, 255, 291, 307
- option column 165–168, 183–184, 198–199, 201, 206–207, 228, 235, 246–247, 249–251
 - option 11-Remove link 249
 - option 3-Copy 246
 - option 4-Delete 184, 207
 - type 3-Copy 165
 - type 4-Delete 168, 201
 - Type option 10-Add link 247, 250
- OS/400 tape device 260

P

- parameter Autostart 220
- PGM 221, 236
- planning a backup strategy 258
- point-to-point (PTP) 305
- Point-to-Point Virtual Ethernet LAN
 - Windows side 282
- point-to-point virtual Ethernet LAN 282, 305
- Primary Domain Controller (PDC) 301
- program product 154, 268–269, 271–275, 283–286, 293
- pull-down menu 129, 157, 229, 232, 242, 245, 252, 254

R

- recommended backup schedule 262
- Redbooks Web site 454
 - Contact us xii
- Remote Control 79–80, 101, 218, 222–223, 229–231, 244, 248
- Remote Deployment Manager (RDM) 53
- Remote Supervisor Adapter
 - Remote Control 230
- Remote Supervisor Adapter (RSA) 7, 14, 230–231
- Remote System
 - CTRL-Q utility 196
- remote system 6–7, 20, 27–30, 34–36, 56, 66–67, 69, 71–74, 77, 79, 87–88, 92–94, 151, 177, 185–193, 196–201, 209, 225, 240, 291, 327, 433–435, 443, 445
 - service processor 20, 28
- Remote system configuration
 - name.Clic k 35
 - object 16, 23–24, 27–28, 34, 66, 70, 151, 185, 426, 433, 443

- remote system configuration 16, 19, 23–24, 27–28, 34–36
- Remote system configuration object 36
- Right click 61–62, 64–66, 72, 75, 95, 98–99, 101–102, 156, 158–160, 162, 185, 192, 202–204, 208, 219, 226, 233–234, 238, 243–246, 251–254, 316, 392, 394, 407, 420, 424, 433, 438, 442–443, 445, 447–448
- right pane 109, 158–160, 162, 172–174, 178–180, 190, 192, 203–204, 222–223, 238, 242–243
- RMTSYS 16, 19–20, 27–28, 151, 156, 185, 187–190, 195–201, 206, 209, 225, 240
- RMTSYS object 20, 185, 195, 333, 338, 342, 404
 - Chap settings 436
- RSA II 14, 141
 - IP address 192.168.70.125 141
- RSTLICPGM LICPGM 140, 284, 286

S

- Save Licensed Program (SAVLICPGM) command 270
- SCSI IP address 32, 34, 39, 372–373, 375
- second port 16, 37, 45
- Secure Sockets Layer (SSL) 51
- select Property 25, 35, 40, 109, 159, 179, 185, 190, 203, 208, 219, 243, 246, 254, 316, 321, 407, 424, 433, 438, 442–443, 445
- serial number 6, 27–28, 66–67, 92–93, 177, 179, 183–185, 192, 196–197, 199
 - Blade center chassis 177
 - management module hardware 28
 - server hardware 27
- Server Message Block (SMB) 300–301
- Service Location Protocol (SLP) 114
- Service Processor
 - configuration 23, 151, 172, 175–185, 190, 196
 - configuration window 177, 180
 - first creation 170
 - internet address 177
 - IP-Address change 181
 - Name 92, 183–184
 - new Password 176
 - Remote Control session 426, 429, 439
 - Right click 177
 - right click 172–174, 178–180, 190
 - successful verification 170–171
 - turn communicates 23
 - user profile list 170–171
 - WEB interface 221–222
- service processor (SP) 5–7, 14, 18–20, 23–24, 27–31, 47–49, 54, 65–66, 74, 82, 88, 92–94, 141, 143, 149, 151–152, 169–185, 190, 192–193, 196, 201, 206, 211, 218, 221, 229–230, 232, 237, 241, 327–329, 426, 429, 436, 439
- Service Processor configuration 151, 172, 175–185, 190
- shutdown timer 87, 90, 276, 287
- single target
 - HBA 356
 - initiator 364
- Small Computer System Interface (SCSI) 13
- SP authentication 93, 175–177
 - Service processor 175–176
- staged backup 258
- Storage path 7, 13, 21, 23–24, 26, 30, 32, 56, 87, 92, 99, 151, 208, 215–216, 231, 241, 254, 327–330, 334, 339, 346–350, 352, 355–357, 359–360, 363–366, 372–373, 376–377, 408, 415–416, 418, 423–425, 438–439, 441
- storage repository 299
- Storage space
 - Asp 87, 90
 - backup tip 267
 - n 273, 275, 284
- storage space 3, 6, 16, 28–30, 38, 78, 87, 90, 149, 152, 156, 216, 219, 241, 246–254, 256–263, 266–268, 271–276, 280–281, 283–292, 326–331, 334, 339, 347, 349–351, 356, 359–361, 363–367, 377–379, 398–400, 405, 407–408, 437–438, 441
 - backup 257–258, 260–263, 266
 - data paths 330–331, 408
 - directory object 267
 - File level recovery 257–258, 282, 289
 - individual files 257, 262, 266
 - integrated Windows servers 262
 - level 256–257, 259, 261, 268
 - level recovery 257–258, 282, 289
 - link 272, 281, 326, 347, 350–351, 356, 359–360, 377, 379, 400, 437
 - memory 38
 - name 267, 283
 - single file 289
 - size 38
- storage space backup 257, 260, 262–263, 266
- Subnet mask 27, 78, 85–86, 89, 94, 141, 143, 158, 163–164, 188, 197–198, 200, 217, 314, 321, 404, 423
- switch module 12, 16, 18, 331, 335–336, 353, 355–357, 359–361, 384, 409
- System Asp 87, 90, 273–275, 284–286
- system Asp
 - Storage space 274
- system drive 6, 29, 58, 86, 89–90, 152, 241–242, 244–247, 249–250, 254, 260, 263, 265–266, 268, 271, 273–275, 277–278, 280–281, 284–289, 291, 365
 - complete copy 291
 - new files 263
 - recent copy 260
 - Right Click 242
 - server boots 6
- system drive hot spare 287
- System i5 1–7, 9–14, 17, 20, 23, 30, 35, 37, 43, 45, 47, 49, 54–55, 57, 77, 84–85, 89–90, 94, 255–266, 268–270, 275–277, 280–282, 286, 288, 291–303, 305–306, 310, 315, 319
 - Access 293, 301
 - Backup 257–258, 261, 302
 - bus 306
 - bus structure 4
 - command line 54
 - configuration 23
 - connection 300–301
 - customer 11
 - device 292
 - disk pool 261

- disk storage 256, 268, 300
- expansion tower 5
- hardware 45
- hardware configuration 295
- i5/OS-centric backup 270
- ifs 299–302
- Integrated Server Support 1
- integrated xSeries hardware 255
- Integration 261, 293, 300, 302
- integration architecture 9–10
- iSCSI 10
- iSCSI network 13–14
- iSCSI SAN implementation 10
- LAN adapter 14
- message 89
- model 14
- models 520,550,570 45
- Navigator 269–270, 280–281, 286, 295, 297, 301
- Navigator component 301
- Navigator window 295, 297
- partition 295, 300
- server 256, 261, 269
- side 77
- storage 58
- tape 257, 259, 293
- tape device 291
- tower 4
- user 3, 10, 90, 269
- users technology 10
- Windows integration 261, 288, 302–303
- System i5 tape
 - device 293
 - drive 291, 293–294
- System i5 tape drive 291–295, 297

T

- Tape Device 259, 269, 291, 293–298
 - tape resource 212
- tape device
 - backup windows 269
 - then right click 295, 297
- tape drive 291–299
 - wide range 293
- target HBA 6–7, 13, 16–18, 21, 23, 26, 327, 330, 332, 334, 337–341, 343–360, 362–365, 367–368, 371–372, 374–375, 377–380, 388–389, 395–396, 399–401, 403–405, 408–409, 415–420, 423–424, 437–439, 441–443
 - NWSH 327
 - storage space 437
 - virtual Ethernet LAN 441
 - virtual Ethernet LANs 363
- target HBAs
 - Blade servers 359, 361
 - iSCSI network 329
 - LAN traffic 348, 350, 352, 357, 359, 361
 - little point 334, 339
 - maximum number 334, 339, 400
 - splitting storage spaces and/or virtual Ethernet LANs 351

- virtual Ethernet LANs 349–350, 358, 360, 362
- workload balancing 360, 396
- xSeries servers 344
- TCP/IP interface 41, 57, 77, 153, 219–220, 232, 269, 281–282, 288, 322
- Tivoli Storage Manager (TSM) 269, 298–299
- TSM architecture
 - TSM administrative client 299
 - TSM backup client 299
 - TSM server 299
- Type choice 78, 84–94, 111, 140, 164, 175–177, 183–185, 197–200, 206–207, 229, 236, 247–248, 250, 290, 423

U

- User drive 267–268, 278, 286–289
- user Id 18, 115, 130, 139–141, 143, 145, 169–174, 176–177, 179, 222, 229–230
- SRVPRC object 170

V

- virtual disc 3, 47, 58–59, 86, 99, 104, 152, 155–156, 242, 244, 246, 251–254, 261, 334, 339, 361, 363–364, 380, 382–383, 385–386, 401, 436–438
 - Heavy IO 47
- virtual disk drive 23–24, 29–30, 38, 366, 438
 - actual disk storage 30
 - multi-path group 24
 - SCSI data flow workload 24
 - SCSI data flows 30
- Virtual Ethernet 2, 6–7, 14, 23–27, 31–33, 37–40, 55–58, 77, 86–87, 89, 91–92, 94, 151–153, 161, 164, 208, 210, 217–218, 231, 254, 305–311, 313–315, 319–323
- Virtual Ethernet LAN
 - Types 307
- virtual Ethernet LAN 26, 37, 153, 218, 281–282, 288, 300, 305–307, 314–315, 319, 321–322, 326, 328, 330–331, 334, 339–341, 344–352, 354–357, 359–361, 363–364, 367, 371–386, 389, 391–396, 400, 402–405, 408–409, 412–414, 433, 437, 441–444, 446–451
 - Cmnxx resources 26
- virtual Ethernet network 40, 152–153, 308–310, 312–313, 315, 321–323
- virtual ethernet resource 25–26, 89, 217, 315, 317–321
- virtual local area network (VLAN) 32
- VRYCFG CFGOBJ 221, 228, 235–236

W

- Web browser 141, 143, 177, 193, 201, 218, 222, 224, 229–231, 241
- web site 44, 108, 113, 139, 154–155, 257–258, 270, 279, 292, 294, 301–303
 - Information Center 257–258
 - PDF file 139
 - service pack 155
- Windows application 37, 263
 - disk activity 37
 - virtual Ethernet LAN activity 37

- Windows backup
 - application 255, 258, 261
 - operation 291
 - requirement 261
 - strategy 292
 - technique 266
 - utility 4, 258–260, 290, 294, 298, 300
- Windows drive
 - complete image 291
- Windows environment 28–29, 44, 50, 115, 257, 261, 266, 275–277, 286, 288, 453
- Windows file 256–257, 259, 262–263, 266, 268, 291, 299–302
 - direct copies 299
 - i5/OS IFS 301
- Windows operating system 3, 23, 55, 212, 256, 260, 263, 268, 314
 - service pack level 212
 - unrecoverable failure 260
- Windows Server 2, 6–7, 13, 23, 27, 29, 32, 34–35, 38, 54, 57, 78, 84–94, 149, 151–152, 218–219, 221, 239–240, 256–269, 271–272, 274, 276, 278–283, 285–286, 289–295, 297, 299–301, 308–309, 314, 328, 330–331, 340, 365–367, 377, 380, 387, 389–393, 395–398, 400, 403, 405–406, 427, 430, 447–448
- Windows server
 - centralized, unattended backup 257
 - iSCSI HBA port 32
- Windows side 57, 152–153, 240, 257, 261, 295, 297, 300, 309, 314
 - file level 261
 - gateway values 314
- WINTAR 260
- WRKNWSSTG command 58, 280–281, 288
- manual deployment 372
- manual deployment works 382
- second initiator HBA 348
- Service Processor 327
- service processor 14
- single initiator HBA 331, 334
- single initiator HBA port 17
- too much bandwidth 344
- Virtual Ethernet traffic 347
- Windows 2003 53
- xSeries system 23–24, 152
- IXA card 23

X

- xSeries 1–7, 9, 11–24, 27–29, 34–35, 37–38, 44–45, 47–48, 50–51, 53–55, 65–66, 77, 82–83, 92–94, 97, 140–141, 149–153, 155, 169–177, 181–183, 185–186, 188, 191–194, 196–197, 200–201, 209–211, 218–219, 221, 223–224, 230–233, 236–237, 240–241, 255–256, 264, 279–280, 292, 301, 325–334, 340–352, 354–355, 357, 362, 368–370, 372–373, 376, 378–385, 389–390, 393, 400, 403, 409–410, 412, 418, 425–427, 429–430, 433, 436–437, 439, 443, 447–449, 454
- xSeries server 1, 3, 7, 11–17, 20–21, 23, 27–28, 34–35, 45, 47–48, 53, 77, 150–151, 155, 193, 201, 236, 256, 279, 292, 301, 308, 327–335, 341–344, 346–349, 351, 357, 369–373, 377–383, 389, 400, 418, 425–426, 429, 436, 439, 447
 - additional initiator HBA 370
 - autodeployment works 372
 - automatic deployment 382
 - bandwidth requirements 344
 - boot port 347
 - CTRL-Q utility 193, 201
 - data path 332
 - general requirements 48
 - initiator HBAs 330
 - iSCSI connection 343

Archived



Redbooks

Implementing Integrated Windows Server through iSCSI to System i5

(1.0" spine)

0.875" x 1.498"

460 <-> 788 pages



Implementing Integrated Windows Server through iSCSI to System i5

Understand how iSCSI support is available in System i5

Learn how to plan and implement Windows servers through iSCSI in System i5

Review the networking components of System i5 iSCSI integration

This IBM Redbook discusses the newest implementation of the Integrated xSeries servers, which utilizes the iSCSI protocol to communicate with the System i5. The iSCSI product offers an additional scalable network connection to the System i5 system bus or the HSL loop. In this IBM Redbook, we discuss:

- ▶ The architecture and scenarios for utilizing this new function
- ▶ Concepts and terminology of iSCSI
- ▶ Planning for the iSCSI environment
- ▶ IBM Director Server and iSCSI
- ▶ Operating iSCSI Servers
- ▶ Backup and Recovery of iSCSI attached servers
- ▶ Virtual Ethernet in an iSCSI environment

This IBM Redbook offers planning and implementation advice and guidance for IBM, IBM Business Partners, and customers.

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