Ghost
User Guide

Ghost

User Guide
Revision 5.1-98 #2

Ghost User Guide
Table of Contents

Table of Contents

Table of Contents

About This Guide

This user guide will assist you when using *Ghost*. It is divided into the following parts:

- ❖ Introduction/Overview
- ❖ Getting Started
- ❖ Scenarios and Operations
- ❖ Setting Up
- ❖ *Ghost* Multicasting
- ❖ *Ghost* Procedures
- ❖ Concepts
- ❖ Glossary
- ❖ Appendices
- ❖ Index

Chapter 1: Introduction/Overview describes *Ghost*'s history, versions, uses and capabilities, and lists its hardware and software requirements.

Chapter 2: Getting Started gives step-by-step procedures for installing/uninstalling and running *Ghost*.

Chapter 3: Scenarios and Operations explains how *Ghost* can be used for various scenarios and operations.

About This Guide

Chapter 4: Setting Up provides useful information on how to set up the various *Ghost* options.

Chapter 5: *Ghost* Multicasting explains Multicast usage to provide an efficient way of replicating hard drives to multiple machines.

Chapter 6: *Ghost* Procedures describe how to manage image files, replication and integrity checking.

Chapter 7: Concepts provides background information about disk drives, partitions, multicasting and other hardware basics and concepts.

The Glossary defines the terminology associated with *Ghost*.

The Appendices contain reference material, including command line switches, troubleshooting and frequently asked questions (FAQs).

The Index gives reference information to make using this guide a really easy experience.

Chapter 1

Introduction/Overview

About *Ghost* Software

*G*eneral *h*ardware *O*riented *S*ystem *t*ransfer

Ghost is the ultimate system installer for fast installation and re-covery of DOS, Windows and other workstations. The capabilities of *Ghost* make it particularly useful whether you have just one ma-chine to configure or are part of an organization with a large number of similarly configured workstations to install.

Ghost is essentially a disk and/or partition copying program. The entire contents of a disk may be copied from one disk to another, or may be copied to a disk image file, and that image file can then be used as a template to create copies of the original disk. *Ghost* also allows these operations to be performed on the partitions of the disk. This means that:

❖ the contents of a partition can be copied to another partition
❖ selected partitions can be copied to an image file and that image file can be used as a template to create copies of the original partitions.

Introduction/Overview

When cloning complete hard drives, procedures such as FDISK and FORMAT are a thing of the past. *Ghost* dynamically partitions and formats a target disk "on the fly", allowing FAT12, FAT16, FAT32, and NTFS partitions to be expanded or contracted to fit the target. The source and target disk may be on the same computer, or the target disk may be on a different computer providing the two computers are connected via a network or LPT ports. The source and target disks may also be different sizes – *Ghost* will adjust the position and size of the target partitions automatically.

Ghost copies every required partition, regardless of type, from the source (disk or image file) to the destination. If the source and destination disks were identical in size and structure, then all that would be needed is a sector by sector copy; but in practice, this is seldom the case. *Ghost* positions each partition or logical drive on the target disk using the same rules as FDISK.

Ghost can also be used to save the contents of a disk or partition to a single disk image file, which can be stored on a Network Server, CDROM, SUPERDISK, JAZ or ZIP drive or other removable media. This file can be used for backup purposes, or for cloning copies of the original disk.

Ghost is designed to run under DOS and comes with a user-friendly GUI interface. It is easily run from a DOS boot diskette or other locations. *Ghost* can handle Win95/WinNT long filenames, NTFS partitions, OS/2 extended attributes and even OS/2 boot manager partitions!

Prior to Win95 there was little need for a utility like *Ghost*. Systems could be installed by simply using the DOS XCOPY command. Windows95 introduced long filenames, and XCOPY could no longer do the job – hence the need for a utility like *Ghost*.

Not only is *Ghost* the fastest way to install Windows95, Windows NT, OS/2 and other operating systems, another of *Ghost*'s capabilities is that it can make complete backups of disks or partitions. *Ghost* even copies 'in-use' system files that are missed by other backup utilities. This makes *Ghost* a perfect choice for disaster recovery operations.

The History of *Ghost* and of Binary Research Limited

Ghost started out as a basic disk-to-disk copying facility with no resizing capabilities. From this, *Ghost*'s great feature list grew to include the addition of partition resizing capabilities, FAT32 support, NTFS, Multicasting, and the new Graphical User Interface now all included in Version 5.0.

Binary Research Limited has been providing the world with quality software since 1986. It is a company of dedicated professionals who take pride in the development of software that is renowned for its robustness and ease of use. The company is based in Auckland, New Zealand, and sells its products globally.

To find out more about *Ghost* and Binary Research Limited, visit the websites:

http://www.ghost.com
http://www.ghostsoft.com

This section intentionally left blank

Introduction/Overview

This section intentionally left blank

This section intentionally left blank

This section intentionally left blank

Technical Support

Most packages include 90 days technical support and one year of free upgrades. (HD Upgrade and DR packages are not eligible for upgrades).

Other Products in the *Ghost* Family of Utilities

Ghost-Walker

This standalone program resolves the problem of duplicate Security Identifiers (SIDs) on cloned NT machines. Purchasers of most Professional *Ghost* packages are now eligible to receive a complimentary copy of *Ghost-Walker*, Binary Research's SID-Changer. This program may be run independently of *Ghost* and ensures that each NT machine on a network has a statistically unique Security Identifier. This will apparently be of great significance when Microsoft ships their long-awaited Windows NT Server version 5.0.

Ghost-Explorer

Another standalone program that complements *Ghost* to provide a full 'file backup and restoration' facility. *Ghost-Explorer* quite simply, allows individual files and/or directories to be restored from an 'image' file. Now it's not necessary to have a third-party backup program for individual file recovery - the every-growing family of *Ghost* utilities is all you need!

Ghost G-Disk

G-Disk is a command line replacement for FDISK and FORMAT and is much quicker for configuring a disk's partition table. G-Disk is command-line driven and provides several enhancements over FDISK.

Ghost **Minimum System Requirements**

This section gives a brief overview of the features of *Ghost*, what *Ghost* supports, and the minimum system requirements needed to run *Ghost*.

Hardware/Software	Minimum	Recommended
Processor	386SX	486 or above
RAM	4 Mb (8 Mb for NTFS)	16 Mb
Operating system	DOS 5.0 or above	DOS 5.0 or above
Monitor	VGA	VGA
Pointing device	No pointing device is required to operate *Ghost*	Microsoft compatible mouse

Additional System Requirements for Transfer Modes

Feature	Minimum Hardware Requirement
Peer to Peer LPT/Parallel Port connection	Laplink or equivalent cable Bi-directional parallel port with EPP or ECP compatibility on each PC
Peer to Peer NetBIOS connection	Ethernet or Token Ring Network Interface card Established Network connection which includes one of the following: - crossover Ethernet cable (pins 1236 --> 3612) - coaxial cable - standard cables with Hub or MAU Additional network software: - LSL - MLID ODI driver - Novell Client drivers - Novell NetBIOS driver (NetBIOS)
SCSI Tape driver	DOS ASPI driver SCSI tape drive Tape media
Multicast	Ethernet or Token Ring NIC Established network connection NIC Packet drivers and/or NDIS 2.01 NIC driver Optional Multicast-enabled router Optional BOOTP/DHCP software
Removable media	Removable media drive and media Media drivers required to use in DOS
CD-ROM usage	CD-ROM Writer CD-ROM Writer Disk creation software
Mapped network volume	Network Interface Card Established network connection DOS Network software client to provide mapped drives.

Ghost Features

Disk-to-Disk Cloning

Ghost allows you to easily clone an entire disk. This option copies every partition, regardless of type, from the source to the destination. If the source and destination disks are not identical in size and structure, *Ghost* automatically positions each partition or logical drive on the target disk using the same rules as FDISK. In addition, if the partition is a FAT12, FAT16, FAT32 or NTFS type (which most PC disks are), *Ghost* allows the target partition to be expanded or contracted.

Individual Partition Cloning

Ghost can be used to make a copy of a single partition or to make an image file containing a copy of one or more disk partitions. This option is the perfect solution for systems with multiple partitions where you may wish to clone individual partitions, but not necessarily the entire drive. It is also perfect for workstations, such as Compaq systems, that keep built-in diagnostic information on a separate partition. The main data partition can be cloned, while leaving the computer-specific diagnostic partition intact.

Multiple OS Support

Ghost can be used to clone DOS, Windows 3.x, Windows95 (including OSR releases), Windows98, Windows NT, UNIX, Linux, OS/2 and Netware volumes.

Long Filename Support

Ghost preserves long filenames on all supported file systems.

Multiple Transfer Methods

Ghost works with many different connection types. Drives can be cloned:

- ❖ internally
- ❖ via removable media
- ❖ via mapped fileserver connection
- ❖ Peer to Peer, using parallel ports or NetBIOS networking,
- ❖ TCP/IP Network multicasting.

Improved! **Multicasting**

Ghost now supports multicasting. This significantly reduces network traffic, while performing concurrent workstation loading. This feature uses TCP/IP multicasting, which makes migrations and rollouts much more efficient by eliminating unnecessary traffic when preparing multiple workstations. *Ghost* Multicasting supports:

- ❖ Ethernet networks
- New! ❖ Token Ring networks
- New! ❖ Image file creation
- ❖ Multicast enabled routers
- ❖ Automatic IP Address selection
- New! ❖ Session start scheduling
- New! ❖ Partition only multicasting
- New! ❖ Multiple sessions on same server system.

High Speed Operation

Ghost can load a workstation from an image file containing both Windows95 and the full installation of Office97 in about seven minutes. Its high speed is part of what makes *Ghost* so powerful!

Dynamic Partitioning and Formatting

Ghost will prepare the target drive and/or partition to be the exact partition type as the source. *Ghost* will also format the drive dynamically. No need to prepare the drive at all. Just configure the drive in the BIOS of the system, and *Ghost* will partition and format the target drive as it is loading the drive and/or partition!

Scalable Partitions/Resizable File Systems

Ghost automatically resizes the target drive FAT12, FAT16, FAT32 and NTFS partitions to the largest size that the file system will handle. It will stop you from resizing a target partition to a size that is larger than the file system allows, preventing accidental oversizing of the partition. *Ghost* even allows you to size greater than, or less than, the original partition size - on the fly!

Introduction/Overview

Image file creation

Complete image files of the entire disk or partition contents can be created to allow later restoration or duplication onto other machines.

Improved! **Image Compression**

Ghost supports image compression while saving an image, resulting in a significant reduction in the storage space required to store the image. Images can be compressed by up to 70% depending on the compression method selected and the contents of the partition.

File Robustness

Dump files are now CRC-checked to detect any corruption. Additionally, verification that a disk replicated by *Ghost* contains the same files as the original can be carried out.

New! **Password Protection of Image Files**

Ghost offers additional security by allowing image files to be password-protected.

New! **Individual File Restoration from Image File**

With *Ghost*'s new partner package *Ghost-Explorer*, files and directories in a *Ghost* image file can be restored.

Removable Media and Media Spanning

With *Ghost*, as well as being able to save to local and network drives, images can be saved and spanned onto removable media devices such as JAZ drives, ZIP drives, CD-ROMs, Superdisk drives, and other removable media. *Ghost* allows a disk or partition image to be split across multiple volumes. If it runs out of space while saving an image, *Ghost* will prompt you to either insert another disk (or other media), or allow you to select an alternate location.

SCSI Tape Support

Ghost images can be saved to and loaded from SCSI tape systems, making it a great disaster recovery solution.

Interactive and Batch Modes

Ghost has both an interactive mode and a batch mode.

22

❖ Interactive mode provides an easy-to-use interface.
❖ Batch mode is useful in repetitive situations – especially when a large number of workstations need to be loaded.

Batch mode allows a system administrator to load a workstation without any user intervention, making *Ghost* perfect for large OS migrations and installation/configuration centers. In batch mode you can split image files into smaller segments as well as run *Ghost* in 'quiet' mode with no display of statistics.

New! **GUI Interface**
A GUI interface has been added to make *Ghost* even easier to use.

Improved! **Small Footprint**
The *Ghost* executable is designed to be small enough that it can be run from a DOS boot diskette. This makes it easy to start the duplication process on the target machine and leaves additional space for drivers to be installed.

Improved! **Reliability**
Ghost now has improved disk geometry detection and reliability.

Improved! **Streamlined Application**
Ghost now requires much less conventional memory than before, removing configuration problems caused by having other hardware drivers loaded.

SID Assignment and Generation
SID assignment and generation is now available with *Ghost*'s partner application, *Ghost Walker*. *Ghost Walker* is a utility for dynamically assigning SIDs on Windows NT systems. This makes it easier to duplicate Windows NT Systems, which require a unique SID for each system on the network.

Ghost **Licensing**

Many of the license options available for *Ghost* are sold on a 'per-seat' basis. What this means is that each and every unique target or destination workstation/drive to which you *Ghost* must be covered by your license. For example, if you wish to use *Ghost* to roll out 400 workstations, you need to purchase a 400-machine license. Or, if you use *Ghost* to take a single hard drive and clone it to other hard drives, each new hard drive must be covered by your license.

A 25-workstation license entitles you to *Ghost* to a total of 25 unique workstations - and no more! A 200-machine license entitles you to *Ghost* to a total of 200 unique workstations, the same 200 workstations every time.

With most licenses, once a workstation is licensed, you may *Ghost* to that workstation repetitively, as often as you wish. In addition, there is no limit to the number of image files you may create.

Other licensing options may be available to meet specific customer requirements. Your local distributor will be able to explain these to you in detail once they have an understanding of how you intend to use *Ghost.*

The Different Uses of *Ghost*

Saving and Loading Images to and from a Fileserver

Your engineers can create a 'model' configuration on a desktop or laptop. Add the third-party software (office software, web browser, etc.), and save the configuration onto a server for later recall. When creating the original model image file that is destined to be used with dissimilar hardware, it is best if you remove the model machine's operating system's software drivers for specific devices such as SCSI devices, network adapter cards, proprietary video cards and sound cards. After the machine has been cloned using *Ghost*, and is started for the first time,

the operating system will do a better job of loading and configuring the hardware drivers. This is because Windows95, WindowsNT and OS/2 Warp might fail to load due to drivers trying to bind to dissimilar configurations on startup. You can add these drivers to the operating system after it loads, or alternatively, we recommend that you include the 'source' files in your model image, so that upon needing drivers and system components, they are already available! (E.g, Window95, "CAB" files, WindowsNT "I386 Directory".)
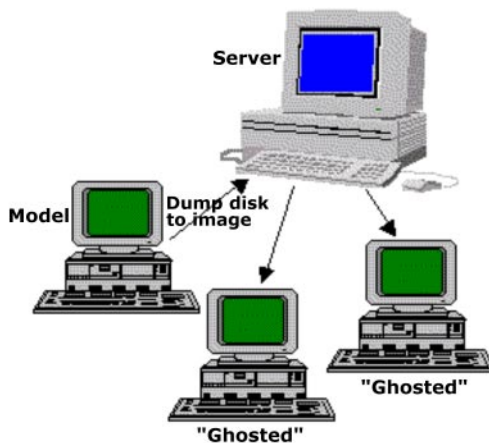
If all machines are identical in hardware then driver conflicts after cloning are less likely to occur.

When restoring the model onto a machine, it is a simple process of booting the machine in DOS with the Network Client installed to provide the mapped volumes where the image is stored and then running *Ghost* to create the image.

Usage: Disk to image file... for saving a hard drive image file to the server
Usage: Load disk from image file... for loading a hard drive from an image file.

Saving and Loading Images to and from CD-ROMs and other Removable Media

It is possible to 'burn' a CD-ROM or to save to a ZIP drive, JAZ drive, Superdisk or other removable media.

For ZIP, JAZ, SuperDisk and removable media *Ghost* will write and read directly 'to and from' the device providing that the device is working and has a drive letter.

When creating the image of the model for storage on a CD-ROM, we recommend bringing the image down onto the PC that has the writing software, and then create the CD-ROM.   Ghost can be included on the created CD-ROM with its functionality limited as outlined in Appendix H.

To create a CD-ROM image file for later restoration:

❖ First, the model workstation is created and the hardware-specific drivers removed to reduce driver conflicts when recreating machines with differing hardware.
❖ Secondly, save your model hard drive to an 'image' file using *Ghost* (onto a network drive, or second drive).
❖ Finally, you would run the CD-ROM writing software and save the image onto the CD-ROM.

For restoration purposes, once the device is working and has a drive letter, *Ghost* can use that drive to perform the required task.

CD-ROM, Optical, JAZ, ZIP, SUPERDISK, etc

Peer to Peer/Slave/Master Overview

Peer-to-peer connections are typically used when there is no network - just two computers that you wish to connect. *Ghost* gives you the choice of connecting via the NetBIOS or via the LPT (printer) parallel port. In both cases, one computer becomes the *master*, the other the *slave*.

❖ To connect via the printer port you need a LapLink or FastLynx type cable plugged into the printer port of both computers.
❖ To connect via NetBIOS you need a network card in each computer, a converted Ethernet cable, and the appropriate networking software. In general, connecting via the network gives two to five times the performance of an LPT connection.

Selecting which PC will be Master

For NetBIOS and LPT connections, one machine must act as the Master and the other the Slave. All operator input will be on the Master computer. Use the following table to choose which will be the Master and which will be the Slave.

Action	Master	Slave
Disk to disk copy	PC containing source disk	PC containing destination disk
Disk to file copy	PC containing source disk	PC receiving destination image file
File to disk copy	PC containing destination disk	PC containing source image file
Partition to partition copy	PC containing source partition	PC containing destination partition
Partition to file copy	PC containing source partition	PC receiving destination image file
File to partition copy	PC containing destination partition	PC containing source image file

Cloning Hard Drives through the NETBIOS Interface

With Ethernet, or Token Ring, you can clone drives between two machines, using their network cards through the NetBIOS protocol interface. By purchasing, or making a custom 'connectivity' crossover cable or using coaxial or a mini-HUB, a PC can connect to another PC. This is a two-node Peer-to-Peer network. This requires five basic Novell networking software components:

- ❖ LSL
- ❖ MLID (the Network Interface Card ODI driver)
- ❖ IPXODI
- ❖ NetBIOS interface
- ❖ NET.CFG (configuration).

These parts loaded with a boot disk, when used with *Ghost*, will allow NetBIOS Master/Slave communication. This will provide the ability to clone between Master and Slave.

Cross-over Ethernet Cable

"Ghosted"
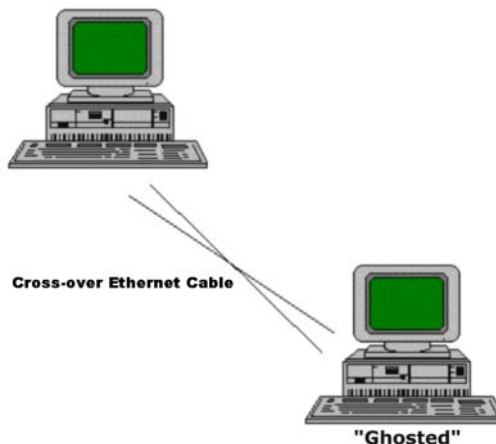
Usage: NetBIOS Master for the machine from which you control the connection. See table on previous page.
Usage: NetBIOS Slave for the other machine participating in the connection See table on previous page.

Cloning Hard Drives through the LPT (Parallel) Port

With two computers and a LapLink cable (not provided with *Ghost*
you can '*Ghost*' Master to Slave!)

Parallel Cable

"Ghosted"
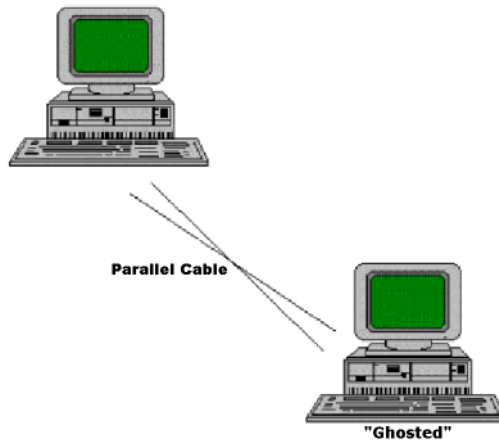
Usage: LPT Master for the machine from which you control the
connection. See table on previous pages.
Usage: LPT Slave for the other machine participating in the connec-
tion. See table on previous pages.

Ghosting Hard Drives Internally

With *Ghost* you can save all of the contents on one internal drive to
another by cloning drive-to-drive. You must have the drive jumpers
properly set and the drives must be configured in the CMOS/BIOS
properly. Like all *Ghost* usage, both the source and destination
must be free from file corruption and physical hard drive problems.
This is the fastest method to *Ghost* a drive and is great for quick
hard drive backups. In the event your main drive fails, you could
remove it from the system, change to the backup drive and be up
and running in no time.

"Ghost" contents of one internal drive to another

Usage: Clone disk to disk… for drive-to-drive *Ghost*ing.

Multicasting

The replication of a model workstation onto many computers can be a time-consuming task. One-to-one connections when replicating a small number of computers is fast and efficient, but as the number of machines increases, the time for the overall completion of the entire replication task increases in proportion to the number of computers being cloned.

When *Ghost* is using a one-to-one approach for transferring information, each of the computer drives being replicated receives its own copy of information, and each of these copies needs to be passed through the same network channel.

30

As the number of replications on the same network increases, the time for overall task completion increases due to multiple copies of information being sent through the common information channel.

Ghost Multicasting uses TCP/IP multicasting, an advanced broadcasting technique. *Ghost* Multicasting supports both Ethernet and Token Ring Networks and clears away the bottleneck of having multiple copies of data being passed through the network. *Ghost* Multicasting now includes support for the multicasting of both disk images and partition images, as well as new automatic multicast server session starting options and image file creation.

A multicasting *session* consists of one server, a single image file, and a group of similar *Ghost* clients requiring the identical disk or partition image. The *session name* is used by *Ghost* clients to indicate the session they are to join and listen to.

Ghost Multicasting client is built into the industry standard **ghost.exe** application software. *Ghost* operates in conjunction with the *Ghost* Multicast Server application to provide a fast and easy way of replicating workstations.

Usage: *Ghost* Multicast server for image creation and distribution.
Usage: *Ghost* for image uploading and receiving cloned images.

Chapter **2**

Getting Started

This chapter describes how to install/unistall *Ghost* and gives a guide to the tips and tricks of navigation. Also included in this chapter is a quick guide to the steps required to perform *Ghost* operations.

Sourcing *Ghost*

Ghost is available in two forms:

❖ an evaluation version
❖ a licensable or licensed version.

The evaluation version will time out at a specified date and is intended for evaluation purposes only. This version is not able to have a license applied to it. Please note that the evaluation version is not shareware, and may have certain features disnabled.

A licensed or licensable version of *Ghost* contains support for all features that are included in the license applied to *Ghost.*

How to Install *Ghost* for the First Time

Once sourced and licensed, the *Ghost* application is a standalone executable that requires little or no installation. The *Ghost* executable file **ghost.exe** can be run from any location in DOS. This means that the installation of *Ghost* can involve the copying of the **ghost.exe** file to any number of user-specified locations. For example, **ghost.exe** can be copied to one of the following:

- ❖ the hard drive for speed and ease of access
- ❖ a 3¼" floppy disk
- ❖ a CD-ROM for portability and convenience
- ❖ a network drive for centralization.

Uninstalling *Ghost*

Uninstallation of *Ghost* involves deleting the single ghost.exe executable file.

Upgrading *Ghost*

Most *Ghost* licenses include one year of free upgrades. To upgrade the version of *Ghost* to the latest version available, visit the *Ghost* software technical support site at http://www.ghosthelp.com. This site includes a list of *Ghost* distributors and a link to their web sites. Alternatively, you may be able to source a *Ghost* upgrade from your local *Ghost* distributor and their web site.

This section intentionally left blank

This section intentionally left blank

Ghost **Usage Overview**

The following section gives a quick overview of the steps involved in using *Ghost* to perform everyday tasks. Please refer to the relevant sections for more details.

Step One
Identify what you want *Ghost* to do
- ❖ Disk duplication?
- ❖ Disk image file creation?
- ❖ Disk creation from Image file?
- ❖ Partition(s) duplication?
- ❖ Partition(s) Image file creation?
- ❖ Partition creation from Image file?
- ❖ How many machines are to be involved?

Step Two
**Select the hardware setup method you will use to perform the operation
with** *Ghost*
Internal disk drives and:

❖ no other devices
❖ other peripheral devices not needing additional setup (e.g, SuperDisk
    or Zip drive)
❖ SCSI tape drive requiring DOS ASPI driver setup
❖ third-party device requiring DOS driver setup
❖ mapped network volume accessibility
❖ peer-to-peer connection using LPT printer port
❖ peer-to-peer connection using NetBIOS and Network Cards con-
    nection
❖ Multicast TCP/IP network.

Step Three
Set up the hardware and system for the method chosen
1. On PCs to be running *Ghost*, ensure all hard drive(s) are installed
    correctly, and the BIOS of the system has been configured and cor-
    rectly displays the valid parameters of the drives.

2. Set up additional drivers required for other devices:

    SCSI Tape Drive requiring DOS driver setup
    Install the SCSI ASPI DOS driver in the config.sys file as outlined
    in SCSI Tape Drive documentation.

    Third Party device requiring DOS driver setup
    Install the DOS driver as outlined in device documentation.

    Mapped network drive
    ❖ Install Network Interface Card (NIC)
    ❖ Connect up cabling
    ❖ Set up NIC using manufacturer's installation program
    ❖ Run NIC test program to check NIC and cabling
    ❖ Install DOS network volume connectivity software
    ❖ Map the network volume to a local drive letter.

Peer to Peer connection: LPT
Using a laplink parallel cable, connect up the two PCs. Decide which machine is the Master and which is the Slave.

Peer to Peer connection: NetBIOS
❖ Install Network Interface Card (NIC)
❖ Connect cabling
❖ Set up NIC using Manufacturer's installation program
❖ Run NIC test program to check NIC and cabling
❖ Install NetBIOS Networking Software
❖ Select which machine is the Master and which is the Slave.

Multicast TCP/IP Network
❖ Install Network Interface Card (NIC)
❖ Connect cabling
❖ Set up NIC using Manufacturer's installation program
❖ Run NIC Test program to check NIC and cabling
❖ Set up Multicasting and perform Multicast operations as detailed in the Multicast procedures section.

3. Test Hardware and DOS driver setup.

Step Four
Start *Ghost* with any additional and optional switches required. Refer to Appendix A for the *Ghost* switch list.

Step Five
Select *Ghost* operation from menu. The menu options are:

❖   Local   ]
❖   LPT   ]   Disk / Partition / Image file / Check
❖   Multicast]
❖   NetBIOS   ]

Step Six
Select source.

Step Seven
Select destination. *Warning*: *Choose carefully.* Ensure that you are overwriting the correct destination. You will most likely be unable to recover from incorrectly selecting the destination drive.

Step Eight
Provide any details required.

Step Nine
Check all details provided are correct and proceed with clone.

Step Ten
Reboot the machine.

Multicasting Step-by-Step Guide

Step One: Set up hardware
- ❖ Install the Network Interface Card (NIC)
- ❖ Connect cabling correctly
- ❖ Set up NIC using manufacturer's installation program
- ❖ Run NIC test program to check NIC and cabling.

Step Two: Decide on TCP/IP and networking settings
- ❖ Networking setup
- ❖ IP address assignment scheme
- ❖ BOOTP/DHCP vs Manual configuration
- ❖ NIC driver selection
- ❖ Overall requirements.

Step Three: Set up *Ghost* Multicast Server
Windows (ghostsrv.exe)
- ❖ Copy **ghostsrv** onto the Windows machine
- ❖ In Windows install a TCP/IP network stack
- ❖ If using Windows 95 as the server, you will need to install the Winsock 2 Update and reboot. (See Appendix C).

DOS (dosghsrv.exe)
- ❖ Copy **dosghsrv.exe** onto the boot disk
- ❖ Create DOS packet driver boot disk as per instructions below.

Step Four: Set up *Ghost* Client (ghost.exe)
- ❖ Copy **ghost.exe** onto the boot disk
- ❖ Create DOS packet driver boot disk as per instructions below.

Step Five: Start the *Ghost* Server.
Start and configure the *Ghost* Multicast server.

Step Six: Start the Ghost Clients
Start the *Ghost* clients and begin multicasting.

Step Seven: Restart
Restart client machines when session is complete.

Setting Up a DOS Packet Driver BOOT Disk

Option One - NDIS drivers with packet driver shim BOOT disk setup
1. Create a floppy boot disk by inserting a blank formatted floppy disk and at a command prompt type
   ```
   sys c: a:
   ```

   Note: You can delete the **drvspace.bin** file from the floppy boot disk to create more room for files.

2. Unzip the **ndisfile.zip** included on your Ghost disk and copy these files to your floppy boot disk:

   autoexec.bat
   config.sys
   dis_pkt.dos

protocol.ini
wattcp.cfg
netbind.com
protman.dos
protman.exe

If you do not have the last three files, they can be sourced from ftp://ftp.microsoft.com/bussys/clients/MSCLIENT/ or in NT4 Server's Network Client Administrator at \\Clients\msclient\netsetup.

3. Copy the NDIS 2.01 driver (**????.dos)** from your network card driver diskette's NDIS directory to this floppy boot disk.

4. Edit the appropriate lines in these files:

CONFIG.SYS
device=????.dos   (change ????.dos to refer to your driver that was copied in Step 3.)
example: `device=elnk3.dos`

PROTOCOL.INI
[PC_CARD]
**drivername=????$**   (change ????$ to refer to your device's name - all UPPERCASE)
example: `drivername=ELNK3$`

WATTCP.CFG
IP=   (change to a unique IP address or leave blank if using DHCP)
NETMASK=255.255.255.0 (change as required for subnet)

Note: You can optionally edit the **autoexec.bat** file to start *Ghost* with switches as required.

5. Copy **ghost.exe** or **dosghsrv.exe** onto the floppy boot disk as required. If there is insufficient space for **ghost.exe** or **dosghsrv.exe** you will need to use a second diskette. Make certain your **wattcp.cfg**

is located in the same directory as the application.

Option 2 - Network card dependent packet driver boot disk setup

1. Create a floppy boot disk by inserting a blank formatted floppy disk and at a command prompt type:
    ```
    sys c: a:
    ```

    Note: You can delete the **drvspace.bin** file from the floppy boot disk to create more room for files.

2. Copy or create these files on your floppy boot disk:
    autoexec.bat
    wattcp.cfg

3. Copy the **????.com** packet driver from your network card driver diskette's PKTDVR or PACKET directory to this floppy boot disk.

4. Create or edit the appropriate lines in these files:

    AUTOEXEC.BAT
    ????.com 0x60   (change ????.com to refer to your
                                    driver that was copied in step 3)
    WATTCP.CFG
    IP=   (change to a unique IP address or
                                    leave blank if using DHCP)
    NETMASK=255.255.255.0 (change as required)

    Note: You can optionally edit **the autoexec.bat** file to start *Ghost* with switches as required.

5. Copy **ghost.exe** or **dosghsrv.exe** onto the boot disk as required. If there is insufficient space for **ghost.exe** or **dosghsrvb.exe** you will need to use a second diskette. Make certain your **wattcp.cfg** is located with it.

Refer to Appendix D "The Wattcp.cfg file" detailing the receive_mode option if multicasting does not work correctly.

Chapter **3**

Scenarios and Operations

Overview

Ghost's range of features and abilities allows it to be used in a wide range of solutions. *Ghost*'s abilities to clone hard drives and partitions provide a flexible and powerful tool that can be used for anything from upgrading the hard drive in your PC at home, right through to managing organization-wide system configuration in large corporations.

In this chapter we introduce a few of the many applications of *Ghost*, and hopefully provide you with an insight into the power and control *Ghost* and its associated utilities offer to carry out the tasks you require. And to top it all off, it's very easy to use!

Ghost is used for:

1. <u>Manufacturers and Integrators</u>
    - One-time installation of software ready to go out the door.
    - CD-ROM and read-only version for Disk Drive recovery.

2. <u>Operating System Rollouts for Companies and Educational Facilities</u>
    - Rolling out a new operating system like Win95 or Win NT to many workstations based on image files created:
        - via Multicasting

           - via network drive
           - other connection methods
           - in conjunction with *Ghost Walker*.

3. <u>Disaster Recovery</u>
   Backup to an image file and complete restoration of partitions and disks
   - diskette/JAZ drive and other removable media
   - network drive
   - second drive
   - restoration of individual files using *Ghost Explorer*
   - classroom restoration on demand.

4. <u>General</u>
   - Hard Disk copy and Hard Drive Upgrades
   - disk to disk internal (upgrading to larger hard drive)
   - LPT, NetBIOS etc.

The following scenarios are just a few examples of when and why you need *Ghost* in real-world situations. In all cases, you are responsible for ensuring that you have the appropriate licenses for all the copies of software that you are installing on each new machine.
<div align="center">***Piracy is a Crime!***</div>

Scenarios and Operations

Scenario 1
Your company has decided to upgrade from Windows 3.11 to Windows NT workstations. You have 25 workstations to configure, and only a day to do it.

With *Ghost*, you can create a model system with all of the necessary software installed (Office Software, Web Browser, etc.), and then save an image of the system to a network server. You can then use *Ghost* to

load the image on to other machines over the network. If you are using *Ghost* Multicast Server, you can load multiple machines at once, dramatically reducing installation time and network traffic. *Ghost Walker* can then be run to provide each Win NT machine with a unique SID, which apparently will be required when NT5.0 Server arrives.

Scenario 2

Windows95 needs to be restored on several problematic workstations, but the workstations are not currently networked.

Ghost allows you to create a drive image on removable media, such as ZIP disks or JAZ disks. You simply create a model system with all of the necessary software installed (as in the preceding example), and then save the drive image directly to the device. *Ghost* will prompt you if it runs out of space on the removable disk and will prompt you to either insert another one or save to an alternate location. Alternatively, an LPT port connection could be used.

Scenario 3

A university provides notebook computers to all new students and would like an easy way for students to restore the computers to their original state if needed.

Ghost can be used to install a complete Windows95 system (or other operating system) from a single disk image file held on a CD-ROM. This technique is currently being exploited by many universities, who issue students a CD-ROM containing a *Ghost* disk image file and *Ghost*. Students can re-load their notebook computers from the CD-ROM at any stage, just by clicking on the *Ghost* icon. No further user-input is required.

'Burning' a *Ghost* disk image file onto a CD-ROM is a useful technique for any organization that distributes updates via CD-ROMs. The version of *Ghost* included on the CD-ROM can be configured to limit the functionality it provides to the end user.

Scenario 4

You want to clone one machine to another machine as quickly as possible, and the machines have network cards.

Ghost allows you to connect two computers directly using NetBIOS. Make sure both computers are connected via a converted Ethernet network cable, or a mini-hub. Both computers must be running the minimum network software for a NetBIOS connection (as in the example below). One computer is then set up as the master and the other as the slave. *Ghost* will automatically connect the two computers.

It may be useful to have a special boot diskette that loads just the network software needed by *Ghost* for the NetBIOS connection.

Example:
For an NE2000 network card the following files would be required (your drivers and protocols might vary - this is only an example).

```
LSL.COM
NE2000.INS
NE2000.COM
IPXODI.COM
NET.CFG
NETBIOS.EXE
```

These following files could be called from the autoexec.bat file which would load the network drivers needed for *Ghost* to work with NetBIOS:

```
LH LSL
LH NE2000
LH IPXODI
LH NETBIOS
```

Scenario 5

You want to clone one machine to another machine using a parallel cable.

Make sure both computers are connected via the printer port with a LapLink/InterLink-type parallel cable. *Ghost* must be running under DOS on both computers. Select Master and Slave computers. The parallel port MUST be set to bi-directional, EPP or ECP (not undirectional). You can experiment with the mode for best performance.

Scenario 6

In your organization, you will be cloning new systems on a regular basis and you want to set up a system that uses the cloning method with the highest speed possible.

Use the *Ghost* disk-to-disk cloning option. Set up a system with one drive configured as the model drive to be cloned (as above). Whenever a new drive needs to be cloned, simply attach the new drive to the controller in the model system, run the drive auto-configure in the BIOS, and then run *Ghost* with the Clone Disk-to-Disk option.

Scenario 7

You need a complete drive backup solution that won't miss 'in-use' system files and can be run from a command line.

Ghost is the perfect solution for making complete backups of disks. It even copies 'in-use' system files that are missed by other backup utilities, making it a perfect choice for disaster recovery operations.

For backups you will generally want to use Batch mode. Batch mode can automate backups, and it allows more control via command line configuration options.

Scenarios and Operations

Scenario 8

You are in charge of several university multimedia labs running Windows95 and need to be able to refresh or update the machines with very little notice. There's only one small problem - the labs are located in the basement, and the main server is on the top floor of the building, and you want to do it as fast as possible!

With *Ghost* Multicasting, you can establish the model configurations of the systems at any stage and save them onto the server machine. When the time comes to restore the labs, you can start up the *Ghost* Multicast Server with the time you would like the session to start, and/or a specific client count. (This configuration could be set up to be run from a third party scheduler application if required.)

Once this has been arranged, the clients can be started using a floppy disk, or alternative boot-up option and run *Ghost*. *Ghost* can then use DHCP to get its TCP/IP configuration details and connect up to the *Ghost* Multicasting session. Once the last PC connects, or the time to start occurs, the Multicast session will start automatically and send the contents of the partition or drive to the lab PC's.

Scenario 9

A System Administrator for a mid-size, full service, law firm maintains 16 Windows NT servers. Backups and archiving of information are saved to JAZ drives. A compressed mirror image of the main NT server was regularly backed up to JAZ cartridges. The main NT server suffered a huge electrical spike as a result of a major lightning storm which overwhelmed the Uninterrupted Power Supply (UPS) system.

Using *Ghost* save a compressed mirror image of the main NT server to JAZ cartridges. Following the lightning strike use the backup image to restore everything to what it was before the disaster. What would normally take more than several, full-time days of work will take only one hour!

Chapter 4

Setting Up

Overview of Setting Up of Each Transfer Method

To work with internal drives you must ensure that each of the drives is properly configured. This means that if fixed IDE drives are in use, the jumpers on the drives have been set up correctly, and the BIOS of the PC has been configured to the disks and setup arrangement. As with all *Ghost* usage, both the source and the destination drives must be free from file corruption and physical hard drive defects.   An overview of the steps required is in the Getting Started chapter.

Local Devices

SCSI Tapes

To use *Ghost* with SCSI tape devices, the tape device needs to have an Advanced SCSI Programming Interface (ASPI) driver for DOS installed. The driver is installed in the **config.sys** file as shown in the example below:

```
device=C:\scsitape\aspi4dos.sys
```

Refer to SCSI Tape Device documentation for further details.

Peer-to-Peer Conncections

Peer-to-peer connections enable *Ghost* to run on two machines and allow *Ghost* to transfer drives, partitions and use image files between them.

Parallel Port

Connect both computers via the printer port with a LapLink/InterLink-type parallel cable. *Ghost* must be running under DOS on both computers. Select which computer is the *Master* (the machine from which you control the connection), and which is the *Slave* (the other machine participating in the connection). All operator input will be on the Master computer. Use the table below to choose which will be the Master and which will be Slave.

Action	Master	Slave
Disk to disk copy	PC containing source disk	PC containing destination disk
Disk to file copy	PC containing source disk	PC receiving destination image file
File to disk copy	PC containing destination disk	PC containing source image file
Partition to partition copy	PC containing source partition	PC containing destination partition
Partition to file copy	PC containing source partition	PC receiving destination image file
File to partition copy	PC containing destination partition	PC containing source image file

The parallel port *must* be set to bi-directional or EPP or ECP, but *not* Uni-directional. You may need to experiment with the mode for best performance.

NetBIOS Network setup

NetBIOS is only available for peer-to-peer *Ghost* usage. Point your Web browser to the *Ghost* Operations page at http://www.ghosthelp.com to download sample files and additional instructions.

Sourcing NetBIOS

The following steps explain the NetBIOS setup procedure.

1. Install network cards.

2. Once the network cards are installed, the two peer *Ghost* machines need to be connected using cabling. The type and setup of the connection will depend on your individual network requirements. These can include converted twisted pair cables, coaxial, hub and MAU-based setups.

3. Run the Network Interface Card setup program and configure card.

4. Set up NetBIOS. An example is shown below. Refer to http://www.ghosthelp.com for files and instructions.

Here is an example of loading NetBIOS in your **startnet.bat** or **autoexec.bat**:

```
LSL.COM
NE2000.COM (replace NE2000.COM with your network card driver)
IPXODI.COM
NETBIOS.EXE
```

An example of the **net.cfg** for an NE2000 NIC:

```
#set up the NIC
link driver NE2000
_int 10
_port 300
```

Setting Up

Select which computer is the *Master* (the machine from which you control the connection), and which is the *Slave* (the other machine participating in the connection). All operator input will be on the Master computer. Use the table below to choose which will be the Master and which will be Slave.

Action	Master	Slave
Disk to disk copy	PC containing source disk	PC containing destination disk
Disk to file copy	PC containing source disk	PC receiving destination image file
File to disk copy	PC containing destination disk	PC containing source image file
Partition to partition copy	PC containing source partition	PC containing destination partition
Partition to file copy	PC containing source partition	PC receiving destination image file
File to partition copy	PC containing destination partition	PC containing source image file

The parallel port *must* be set to bi-directional or EPP or ECP, but *not* Uni-directional. You may need to experiment with the mode for best performance.

Mapped Network MS Windows Volume

How to create an NT Server 4.0 Network Boot Disk for DOS

In order for *Ghost* to execute reliably, you can't have the Graphical
OS up and running due to contention issues. Since the Windows op-
erating system is not running we have lost access to our network, so
what we need to do is create a network boot disk. A network boot disk
will contain the appropriate network drivers to give us the ability to
connect to a network.   Point your Web browser to the *Ghost* Opera-
tion page at http://www.ghosthelp.com to download sample Microsoft
Client files.

Create a System Disk

Since NT doesn't allow you to create a system (boot) disk, have Win-
dows95 do it for you. Follow these quick steps:

1.   Double click on the *My Computer* icon and right click on the
     floppy drive and choose *Format*

2.   Choose Copy System Files, this will create the bare minimum
     Boot Disk.

3.   Now we need to add the network files.

Add Networking Files

Now that we have our System Boot disk we need to add the network-
ing files to allow us to access the server. Follow these steps to have
NT Server add the network files required.

1.   Go into Administrative Tools on NT Server and start the Net-
     work Client Administrator.

2.   Choose the first option - Make Network Installation Startup Disk.
     This option will set up our disk to access the Client Installation
     for NT Server. We don't really want to access these installation
     files but this process will configure the components so that we

can connect to the server. Essentially, this is all we need to accomplish.

3. Now choose the location of the Client Installation files (see screen below). This is done either by setting up the NT Server CD Client directory as a share, or by copying the NT Server CD Client directory onto the server and setting up this directory as a share.

4. Choose Network Client v3.0 (see screen at top of next page). You must also choose your Network Card. If your network card is not on the list, temporarily pick something similar.

Target Workstation Configuration

Select the options for the target workstation.

Floppy Drive

( ) Drive A: is 3.5"
( ) Drive A: is 5.25"

Network Client:

Network Client v3.0 for MS-DOS and Windows
Windows95

Network Adapter Card:

3Com EtherLink

OK
Cancel
Help
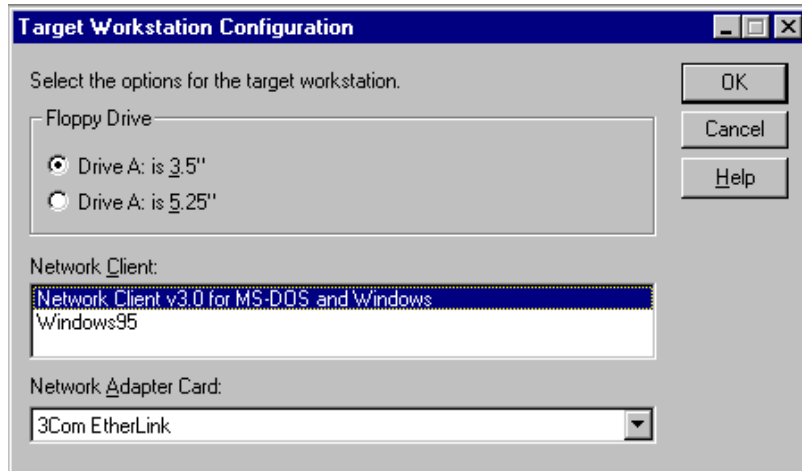
5.   To log on to the NT server you need to provide some basic infor-
     mation needed for NT networking (see screen below).

Network Startup Disk Configuration

Select the options to be used by the network startup disk.
These options only apply during the startup process.

Computer Name: GHOST
User Name: Administrator
Domain: PROGRAMMING
Network Protocol: NetBEUI Protocol

TCP/IP Settings

☑ Enable Automatic DHCP Configuration

IP Address: 0.0.0.0
Subnet Mask: 0.0.0.0
Default Gateway: 0.0.0.0

Destination Path: A:\

OK
Cancel
Help

6. Make sure that the system disk that you created is in the drive, NT will add all of the networking features at this point.

7. At this point you have your complete network boot disk with a few modifications to make. We want to log into the network and map a drive. Make changes to your **autoexec.bat** file to be as shown below:

   ```
   path=a:\net
   a:\net\net start
   net use z:  \\KERMIT\images
   ```
   (change the \\KERMIT\\ argument to your \\server\share location)

8. Once this has been completed you are set, unless the network card that you selected in Step 4 was incorrect. If that's the case go on to the next step. If you think you have the right card but the network boot disk gives you an error message about binding, there is a problem with the card configuration.

9. The following explains how to add an NDIS driver for NIC cards that were not originally included in the Network Client Administrator. This is an excerpt from article Q142857 (How to Create a Network Installation Boot Disk) in the Microsoft Knowledge Base.

This example is for the Intel EtherExpressPro NIC card; however, this example can serve as a template for all NDIS compatible drivers.

- Copy the **epro.dos** driver to the A:\Net directory.
- Modify the **net\protocol.ini** file and change the [EPRO$] section to `drivername=EPRO$`.

   NOTE: The "drivername=" information is included on most manufacturers' disks in the sample **protocol.ini** file that comes with the network card.

- Modify the **net\system.ini** file and change the [network drivers] section to `netcard=Epro.dos`

Note: The file that the "netcard=" equals is also on the manufacturer's disk.

Additional drivers can be found on the Windows NT Server compact disc in the Clients\Wdl\Update directory. For more information, please see article Q128800 (How to Provide Additional NDIS2 Drivers for Network Client 3.0) in the Microsoft Knowledge Base. Additional information is available at www.ghosthelp.com.

You are now finished. If you get an error message stating an inability to bind, the network card is not configured properly. Check your interrupt and address settings on your card and in your protocol.ini. Also make certain you have customized properly as explained in Step 10.

Ghost Multicasting

Multicasting allows multiple *Ghost* machines to receive the same information over a computer network simultaneously through the use of a single transmission of the information. Two applications are used in *Ghost* Multicasting, one at the server, and another at every client workstation to be cloned. Point your Web browser to the *Ghost* Operation page at http://www.ghosthelp.com to download sample Multicast clients.

At the server:
Ghost Multicast server – The Windows version **ghostsrv.exe**, or the DOS version **dosghsrv.exe** – that distributes the image file using TCP/IP Multicasting.

Important: If using Windows95 as the server you will need to in-

stall the Winsock2 update that is available at http://www.ghosthelp.com/downloads/multidl/ws2setup.zip

At a workstation:

Ghost Multicast client – The DOS application **ghost.exe**, receives and writes the image file to the local disk drive.

Windows applications TCP/IP settings are assigned in the Windows network settings. DOS applications TCP/IP settings are customized in the **wattcp.cfg** configuration file and require a packet driver interface to be set up.

IP Addresses and *Ghost* Multicasting

Each computer on an IP network needs a unique IP address. Associated with an IP address is a subnet mask. The subnet mask indicates the range of IP addresses that are locally accessible to the computer. Each of these locally accessible computers becomes a member of the local subnet. If the address of another computer is outside the range of IP addresses specified by the subnet mask, then this computer is known to be on a different subnet.

To communicate with a computer on another subnet, the local computer sends the information to the 'default gateway'. It is the default gateway's responsibility to forward information to the correct receiver. The default gateway of a computer needs to be on the same subnet as that computer. These values can be:

❖ specified locally on a computer in a configuration file, or,
❖ automatically using a BOOTP or DHCP system.

Allowed addresses for internal private internets (RFC1597)	
10.0.0.0 - 10.255.255.255	(1 class A)
172.16.0.0 - 172.31.255.255	(16 class B addresses)
192.168.0.0 - 192.168.255.255	(255 Class C address)

Locally specified IP address

An IP Network using locally specified addresses requires each machine to be manually set up to have:

❖ A unique IP address
❖ The correct subnet mask
❖ The default gateway (optional)

The Windows *Ghost* Multicast Server receives its locally specified IP address, subnet mask and default gateway from the TCP/IP parameters in the Network option of the control panel in Windows.

The DOS-based *Ghost* Multicast Server and Client(s) receive their IP address, subnet mask and default gateway from the configuration file named **wattcp.cfg** that is usually located in the same directory as the *Ghost* executable file.

Note: If a DOS boot disk is used to start up *Ghost* Multicasting, with locally specified IP addresses, each PC requires a different wattcp.cfg file to be specified and therefore every boot disk for the workstations is unique.

Example of Client Configuration Files wattcp.cfg

Windows95 PC #1 running Windows *Ghost* Multicast Server (ghostsrv.exe)

IP address   :   192.168.100.10
Subnet mask   :   255.255.255.0
Default gateway: 192.168.100.1

Uses Windows tcp stack configuration so there is no need for **wattcp.cfg** file.

DOS PC #2 running *Ghost* client (ghost.exe)
IP address   :   192.168.100.3
Subnet mask   :   255.255.255.0
Default gateway: 192.168.100.1

DOS PC #2 wattcp.cfg file is as follows:
IP = 192.168.100.3
NETMASK = 255.255.255.0
GATEWAY = 192.168.100.1

DOS PC #3 running *Ghost* client (ghost.exe)
IP address   :   192.168.100.44
Subnet mask   :   255.255.255.0
Default gateway: 192.168.100.1

DOS PC #3 wattcp.cfg file is as follows:
IP = 192.168.100.44
NETMASK - 255.255.255.0
GATEWAY = 192.168.100.1

Any address other than 192.168.100.0 to 192.168.100.255 is on another subnet and needs to be passed on to the default gateway (192.168.100.1 in this example).

If the computers do not need to communicate with another machine outside their subnet, then a default gateway is **not** required.

Automatically-generated IP address using DHCP/BOOTP
If a BOOTP or DHCP server is installed on the network, one may take advantage of Dynamic Host Configuration Protocol (DHCP) or BOOTP for IP address assignment. A DHCP server is included in Windows NT Server release 4.0. Other DHCP and BOOTP applications are available for various operating systems and can be used with *Ghost* Multicasting.

For multicasting to many clients, the advantage of not having to edit a unique **wattcp.cfg** file on every client may well be advantageous. Balanced against this is the additional complexity of the DHCP setup. Refer to the section "BOOTP/DHCP Automatically Defined IP Address" later in this chapter.

Example of locally-specified IP address for Windows95

Note: If the Network Administrator provides default addresses, these addresses should be entered instead of the addresses in the following example.
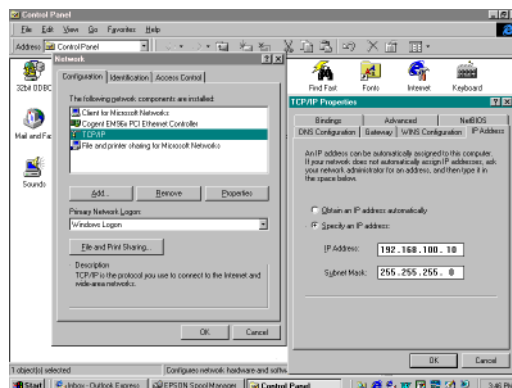
1. Double click on Desktop | My Computer | Control Panel | Network.

2. Confirm that the network TCP/IP entry is selected and double click that entry.   Result = The TCP/IP properties dialog box displays.

3. Select the 'IP address' tab and enter the numerical values supplied by the system administrator, or if no pre-defined values are available, use these default values:
    IP address   192.168.100.10
    Subnet mask   255.255.255.0

4.   If a router is being used, select the 'Gateway' tab and enter the appropriate IP address, in this example 192.168.100.1. Otherwise leave this entry field empty.

The above procedure has assigned default values for the example Windows95 Multicast Server machine that will be used in this document.

BOOTP/DHCP Automatically Defined IP Address

Specifying a local configuration for every computer on an IP network is sometimes inconvenient, or impractical. *Ghost* Multicasting supports the automatic, or remote definition of IP addresses and network parameters using BOOTP and DHCP systems.

To be able to use BOOTP or DHCP to specify a computer's IP address requires the running of a BOOTP or DHCP server on the network. This BOOTP/DHCP server listens on the network for computers requesting an IP address to be provided, and replies with the address the DHCP/BOOTP server is configured to provide. The BOOTP/DHCP server needs to be configured to provide the IP address, Subnet mask, and (optionally) the default gateway.

Examples:

Windows NT 4.0 server #1 running Windows *Ghost* Multicast server ghostsrv.exe and Windows DHCP server
IP address   :   172.16.5.10
Subnet mask   :   255.255.255.0
Default gateway: 172.16.5.1

DOS PC #2 running *Ghost* client ghost.exe
IP address   :   supplied via DHCP
Subnet mask   :   supplied via DHCP
Default gateway: supplied via DHCP

(DOS PC#2 **wattcp.cfg** file is *empty or does not exist* since *Ghost* Multicasting defaults to using BOOTP and DHCP if no local specified network parameters are supplied).

DOS PC#3 running *Ghost* client ghost.exe
IP address   :   supplied via DHCP
Subnet mask   :   supplied via DHCP
Default gateway: supplied via DHCP

(DOS PC#3 **wattcp.cfg** file is *empty or does not exist* since *Ghost* Multicasting defaults to using BOOTP and DHCP if no local specified network parameters are supplied.)

The controlling element for DHCP is the DHCP server that serves the requests of clients and ensures no duplicate IP addresses exist on the network. Since many DHCP servers can be placed on a network, active measures must be taken to avoid duplicate address generation and its attendant problems.

Setting Up DOS-based Multicast Client and Server

DOS Client boot disk creation

The following steps are required to create the floppy disk:

1.   Transfer the system files onto a blank diskette to create a bootable floppy.

2.   Set up the Packet Driver Interface to the Network Interface Card (NIC).

3.   Copy **ghost.exe** and **wattcp.cfg** to the floppy disk.

4.   Edit the **wattcp.cfg** file to set the IP address and Netmask address.

63

5.   Edit the **autoexec.bat** file to start the **ghost.exe** application and join a Multicasting session.

Step 1 - Copy system files onto a blank floppy disk
Insert a blank formatted floppy disk into A: drive and transfer the system files by either of the following commands:
```
C:\>  sys c: a:
C:\>  format a: /s
```

Step 2 - Set up the Packet Driver Interface
Follow the instructions in the Packet Driver Setup section. Ensure all files are installed on the disk. For example, with network card dependent packet drivers, only one file, the packet driver, will be required to be copied onto the disk.

Example: `c:\>  copy ne2000pd.com a:\`

Step 3 - Copy ghost.exe and wattcp.cfg to the floppy disk
Copy the *Ghost* executable and **wattcp.cfg** configuration file onto the floppy disk. Optionally, copy a *Ghost* Batch Switch file onto the disk as required.

Step 4 - Edit the wattcp.cfg file
The **wattcp.cfg** file stores the TCP/IP stack configuration details and specifies the IP address and subnet mask of the machine.

Sample **wattcp.cfg** file:
IP   = 192.168.100.44
NETMASK = 255.255.255.0

See Appendix D for a detailed description of **wattcp.cfg** file configuration keywords.

Step 5 - Edit the autoexec.bat startup file as required
Example: ne2000pd.com 0x60 10 0x280
                ghost.exe

Additional *Ghost* command line switches can be added to automate the *Ghost*ing process. Refer to the Multicast Client command line switches in Appendix A for more details.

DOS Packet driver setup

The DOS-based *Ghost* Multicast Client and DOS-based *Ghost* Multicast Server require an Ethernet-based or Token Ring-based packet driver to be loaded prior to running. The Windows version of the *Ghost* Multicast Server does *not* require a packet driver as it uses the standard TCP/IP windows network support.

The Network Interface Card (NIC) specific Packet Driver setup is relatively simple to implement. If manual IP specification is being used, it requires a customized configuration file on each workstation (**wattcp.cfg**) to set the workstations IP address and subnet mask.

The NDIS Driver setup is certainly more complex to set up than the Packet Driver, and as with NIC specific packet drivers, each workstation using manual IP specification requires a customized **wattcp.cfg** file.

The selection of NDIS 2.01 and Shim, or a Network card specific Packet driver will depend on factors such as:

❖ availability of driver
❖ reliability
❖ ease of use
❖ speed.

By running a system test you will be able to choose the best alternative for your NIC, i.e, specific Packet Driver or the NDIS 2.01 driver and Shim.

Note: *Do not use the network client administrator from Windows NT4 or the Microsoft Network Client Installation program to create a Ghost Multicast boot disk as they are not compatible.*

There are several options available to install the packet driver interface. The option selected will depend on your individual requirements and resources.

Options:

1.   Network Card dependent Packet Driver.

2.   NDIS version 2.01 driver with supplied Packet Driver Shim.

3.   Third Party NIC Driver and Packet Driver Shim. These have not been tested or documented with *Ghost*'s Multicasting feature. This includes ODI-based packet driver shim like **odipkt.com**.

Option 1:
Network card dependent Packet Driver
(also known as Packet Drivers, Clarkson Packet Drivers and Crynwr Packet Drivers).
Packet Drivers have the advantage of being relatively quick to set up and require minimal configuration.

Steps in using a card specific packet driver:

1.   Locate Packet Driver designed for your network card.

2.   Start Packet Driver.

Step 1 - Locate Packet Driver designed for your network card
Packet Drivers are usually supplied on the installation disk included with a network card and usually have a **.com** filename extension. Alternatively, packet drivers may be available from your network card manufacturer's web site.

Step 2 - Start Packet Driver
Follow the instructions included with the packet driver and install it.

Examples:
The command line arguments vary slightly from driver to driver. The variation is due to the fact that each board configuration varies slightly.

3Com 590 PCI network card packet driver:
```
A:\>  3c59xpd.com
```

3Com509 ISA network card packet driver:
```
a:\>  3c5x9pd.com 0x60
```

NE2000 compatible using software interrupt 0x60 at IRQ10 and IObase 0x280
```
a:\>  ne2000pd.com 0x60 10 0x280
```

The syntax of the ne2000pd command is an example of an average ISA driver command line. The IRQ and IO base address values can be found using the setup program included with the network card and the software interrupt can be between 0x60 - 0x7f.

Option 2:
NDIS 2.01 NIC Driver with supplied Packet Driver Shim
Steps required:

1.   Locate NDIS 2.01 driver for network card.

2.   Copy and modify supplied **protocol.ini**, **config.sys** and **autoexec.bat** files. These sample files are provided within **ndisfile.zip** on the _Ghost_ diskette.

3.   Copy other required files.

4.   Reboot your machine to load the stack.

Step 1 - Locate NDIS 2.01 driver for network card
NDIS (version2.01) drivers are usually supplied on the installation disk included with a network card and usually have a **.dos** filename extension. Alternatively, NDIS (version 2.01) drivers may be available from the network card manufacturer's web site.

Step 2 - Copy and modify protocol.ini, config.sys, and autoexec.bat
Base configuration files ready for editing are included in the **ndisfile.zip** file included with the *Ghost* Multicasting files. Extract these configuration files and edit as shown below.

Sample **protocol.ini** file:
```
[PROTMAN]
drivername = PROTMAN$

[PKTDRV]
drivername = PKTDRV$
bindings = PC_CARD
intvec = 0x60
chainvec = 0x66

[PC_CARD]
drivername = PNPND$
```

Notes:

❖ The [PC_CARD] module drivername should be changed to correspond to the NDIS driver in use for your network card. For example if a 3Com 509 card is used then the change required would be:

    drivername = ELNK3$

❖ Any additional required options for the network card configuration can be entered in the [PC_CARD] module. Refer to the documentation or example **protocol.ini** of the NDIS driver for the network

68

card in use if required. For example, the 3Com 509 card allows you to optionally specify the IO Base
address:

```
[PC_CARD]
drivername = ELNK3$
IOADDRESS = 0x300
```

Sample **config.sys** file:
```
device=protman.dos /I:\  ;Required NDIS Driver
device=dis_pkt.dos       ;Shim supplied with Ghost
devide= pnpnd.dos        ;Driver specific to NIC,
                         ;if 3COM509 used, replace
                         ;with device=ELNK3.DOS
```
Notes:

The /I: indicates the **location** of the **protocol.ini** file and must be present. For example: /I:\ **specifies root** directory and /I:A:\NET specifies **A:\NET.**

Sample **autoexec.bat** file:

```
prompt $p$g
netbind
```

Notes:

NETBIND binds the NDIS drivers together and installs the packet driver interface.

Step 3 - Copy other required files
Locate and copy the following files:

protman.dos	**dis_pkt.dos**
protman.exe	**netbind.com**

dis_pkt.dos is included in the **ndisfile.zip** file included with *Ghost* Multicasting. Netbind and protman files can be sourced from MS Network Client 3.0 that is included in Windows NT4.0 server and is downloadable from:
ftp://ftp.microsoft.com/bussys/clients/msclient/ if not locally available.

Please note that boot disks created automatically using MS Network Client will not work with Ghost Multicasting.

Your directory, or floppy disk, should now contain the following files:

System Files	**Config Files**	**NDIS Files**
command.com	config.sys	dis_pkt.dos
msdos.sys (hidden)	autoexec.bat	netbind.com
io.sys (hidden)	protocol.ini	protman.dos
drvspace.bin[1] (hidden)		protman.exe[2]
		???.dos[3]

Notes:

[1] To provide more space on the boot disk **drvspace.bin** can be deleted

[2] **protman.exe** is used during the NETBIND and does not need to be included in the **autoexec.bat** file.

[3] Network card specific driver e.g, ELNK3.DOS.

Step 4 - Reboot your machine to load the stack

Reboot the machine using the configuration files created. The Packet Driver Interface should now be ready for use.

Chapter **5**

Ghost **Multicasting**

Overview

Ghost Multicasting technology allows you to be more efficient with migrations and rollouts when peparing multiple workstations, by almost eliminating replicated network traffic.

This chapter will lead you through the steps needed to use *Ghost* Multicasting. The steps are as follows:

1. Complete network setup requirements.
2. Creation of an image file.
3. Setting up and starting the Multicast server application to send the file.
4. Starting *Ghost* on the workstations to be updated.

Usage of the *Ghost* Multicasting feature can be achieved in several ways:

❖   through the Graphical User Interface (GUI)
❖   Command line switches
❖   Batch files.

or a combination of the above.

Deploying the Windows Multicast Server (ghostsrv.exe)

The *Ghost* Multicast Server Windows application, **ghostsrv.exe** multicasts a copy of an image file to one or more *Ghost* clients listening to the server's session. As well as this, the *Ghost* Multicast Server has the ability to create an image file based on a connected client *Ghost* machine.   A *session* consists of one server, a single image file and a group of similar *Ghost* clients requiring the identical disk or partition image. The *session name* acts as a key, identifies the session and must be unique. The session name is used by *Ghost* clients to indicate the session they are to join and listen to.

The server runs under Windows NT4, Windows95 and Windows 98, and requires the network settings of Windows to include a valid TCP/IP stack.

Note: If you are using Windows95 as the server you will need to install the winsock2 update that is available at <u>www.ghosthelp.com/downloads/</u> <u>multidl/ws2setup.zip</u>.   Do this before using **ghostsrv.exe**. See Appendix C.

The steps are:

1.   Enter a session name and type of session for the Multicast session.

2.   Select the image file and partition details to be used in the Multicast session.

3.   Specify optional Auto-Start parameters.

4.   Press the "Accept Clients" button to start listening for client *Ghost* machines, and accept them into the session.

5.   Start all *Ghost* client machines.

6.   Press the "Send" button to start sending the image file to the

client *Ghost* machines, or allow Autostart feature to commence the session.

Step 1 - Enter a session name

Enter a session name for the Multicast session and specify the type of session. A multicast session name can be any alphanumeric sequence of characters. Spaces are accepted in Graphical Mode but may not be used in command line switches. The case of characters is ignored. Select the desired transfer option. There are two transfer options available:

❖ Load to Clients -   Multicasts out from the server a copy of the selected partition or complete disk image to all connected *Ghost* Multicast client machines in the session.

❖ Dump from Client - Uploads and creates an image file in the location specified in Step 2, "Select the image file", from the first client *Ghost* machine to connect to the created session. This session is only allowed to consist of one *Ghost* Server and one *Ghost* client.
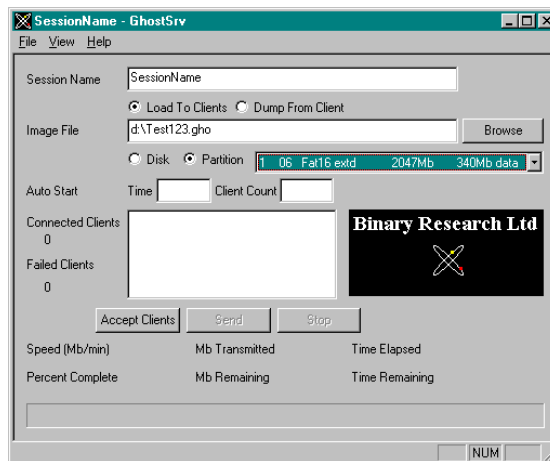
Step 2 - Selecting the image file to be used in the Multicast session
Enter the full path of the disk image file in the Image File Box. The Browse button may be used to locate the file. (See screen on previous page.)

Step 2a - Dump from client - Creating an image file
The image file will be created at the location and filename specified. If the file already exists, **ghostsvr** prompts you if you wish to overwrite. Either the entire disk image (or an image including selective partitions) can be created. For the entire disk image, select the disk option.

If you require the ability to select which partitions on the *Ghost* Model Client are to be included in the image file, select the partition option. This will allow the selection of which partitions on the Client *Ghost* machine are to be included in the image file.

Step 2b - Load from client - Sending an image file
Either the entire disk image or a selective partition image can be multicasted to the Client *Ghost* machines. For the entire disk image, select the disk option.

If you wish to multicast a single partition from an image file, select the partition option and select the partition from the image file. The Disk or Partition settings must be selected before Step 4, "Start Accepting", can be completed.

If the file selected is not a valid image file, Step 4 "Start Accepting", will not be able to be completed.

Note: The *Ghost* Image file can be created by **ghost.exe** through the use of the "Disk to image file" option in either local mode, or using the Multicast "Dump From Client Mode".

Step 3 - Specify optional Auto-Start parameters
The server can be optionally set up to start sending to clients connected to the session automatically. The start time can be based on:

❖ A specified time within the next 24-hour time period

74

❖ The number of clients connected to the session
❖ A logical 'OR' combination of the above two criteria.

The Auto-Start Time box allows a time between 00:00 and 23:59 to be specified. The format of the time is hours:minutes. For example 5:30am would be 05:30, and 5:30pm would be 17:30.

The Auto-Start Client Count box allows the threshold number of joined clients to be specified. For example, if the threshold was set to ten (10), then the server would wait and accept clients until the tenth client. Once the tenth client is accepted, the server would stop accepting any more clients and start sending out to the connected *Ghost* machines.

Logical combination of both criteria:
When both the auto-start time and client count are specified, the first 'TRUE' criterion specifies the start time. For example, if the start time is 17:30 and the client count is 10, then the server will start the session when the 10th client joins 'OR' if the time is 17:30. There must be one or more clients logged to the session for the time option to operate.

These Auto-Start values become active when Step 4, 'Start accepting clients' is activated. Auto-Start can be overridden either by using the stop button, or by starting the 'Send' session earlier using the send button. See screen below:
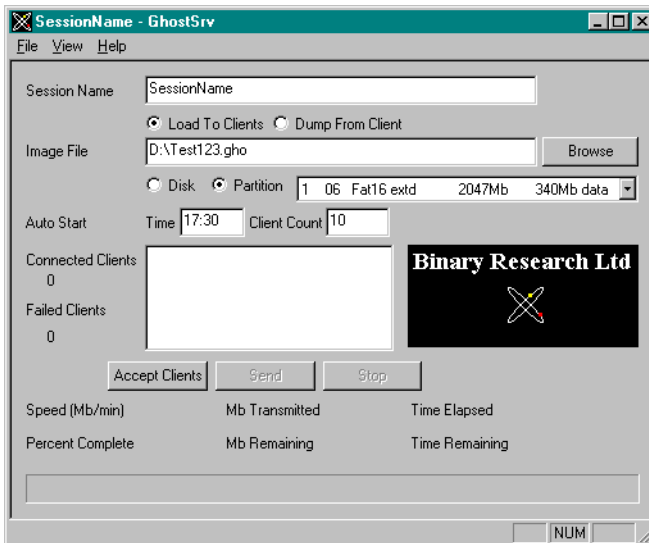
Step 4 - Start accepting the client *Ghost* machines into the session

The Accept Clients button will become active following the completion of Steps 1 and 2. Check that the following are correct:

❖ session name
❖ transfer option (load vs. dump)
❖ disk image filename
❖ image file disk or partition selection
❖ optional auto-start details.

Press 'Accept Clients'. The type of the file is checked and then the server starts listening for clients on the network that are requesting to join the session.

If the server is set up to dump from client, the server will accept the first client to join the session as the model *Ghost* machine and start automatically.

The Auto-start parameters become active once this step has been completed. To override the Auto-start parameters, press the Start or Stop button as required. (See screen below.)
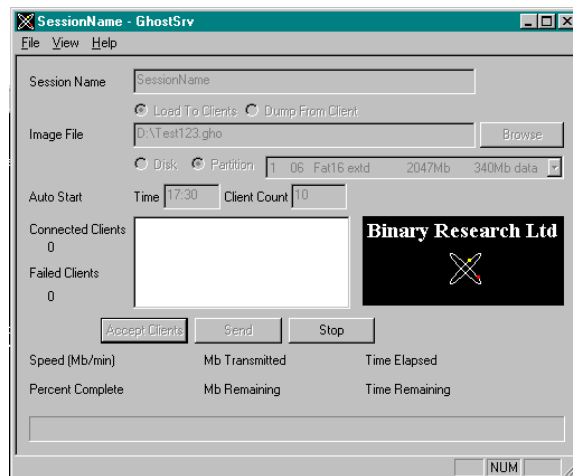
Step 5 - The *Ghost* clients should now be started
The IP addresses of the client *Ghost* machines that are connected and waiting for the Multicast session to start will appear in the 'connected window'. Refer to "Deploying the MultiCast *Ghost* Client" later in this chapter for more details.

Step 6 - Start sending the image file to the client *Ghost* machines
When all *Ghost* clients have connected and are waiting for the Multicast session to start, and the auto-start option is not required, press the Send button. This will start the Multicast transmission of the disk image file. (See screen below.)
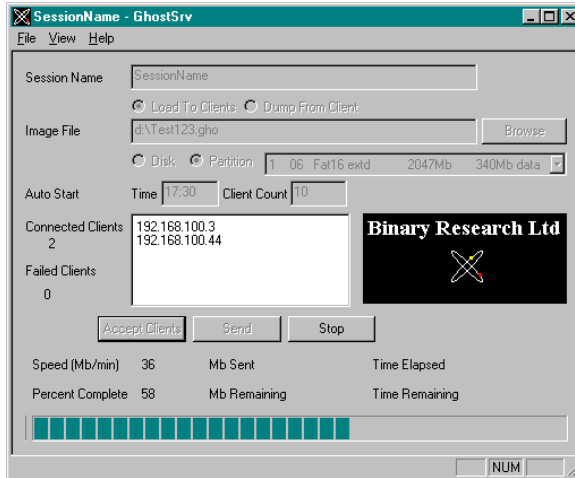
The Progress Indicator will show the status of the Multicast session as it proceeds, along with other image file transfer details. Please note that the statistics shown are based on the image file size, and do not represent the internal compressed data of the image file. The speed shows the actual amount of data being sent over the network in Megabytes per minute from the image file.

Ghost Windows Multicast Server Command Line Options

The Windows-based *Ghost* Multicast server can be run from the command line enabling it to be started using a batch file or third party scheduler application.

The command line interface is as follows:

```
ghostsrv filename session [-nclient_count][-ttime][-
l{A|I|W|S|E}][-flogfile][-c][-d][-p]
```

where:

| -filename | = | path and filename of disk image file to be dis-tributed |
| session | = | session name to use |

and optionally:

-n	=	starts sending to session automatically after client-count clients have joined the session
-t	=	starts sending to session automatically after specified time (24 hour hh:mm format)
-l	=	creates log file with log level specified (E,S,W,I, or A)
-f	=	specifies the logfile name, (by default "**ghostlog.txt**")
-c	=	closedown **ghostsrv** after Multicast session comple-tion
-d	=	use dump from client mode (load to client is default)
-p	=	specifies partition mode operation. If loading to cli-ents, the partition number must be given. If dump-ing from client no partition number is required.

Following are some examples of usage:

Example 1:
Uploading a complete disk image file from a client *Ghost* machine.

```
ghostsrv c:\test123.gho TestSession -d
```

Starts a Multicast session called TestSession and creates or overwrites the file c:\test123.gho. The first connecting client's IP address will be displayed on screen, and the session will start automatically. The Client PC indicates the drive to use for the image creation.

Example 2:

Uploading partitions from a client *Ghost* machine to an image file:

```
ghostsrv c:\test123.gho TestSession -d -p
```

Starts a multicast session called TestSession and creates or overwrites the file **c:\test123.gho**. The first connecting client's IP address will be displayed on screen, and the session will start automatically. The Client PC indicates the drive and partitions to be included in the image created.

Example 3:

General Partition Use:

```
ghostsrv c:\test123.gho TestSession -p2
```

Starts the ghostsrv application and configures a Multicast session called TestSession and uses the second partition in the file **c:\test123.gho**. Once the Accept button has been pressed, the connecting clients IP address will be displayed on screen. Once all Clients have connected, pressing the Start button will commence the session transmission.

Example 4:

Number of clients auto-start

```
ghostsrv -n10 c:\test123.gho TestSession
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. Once 10 clients have connected, the session transmission is started automatically.

Example5:

Time-based auto-start

```
ghostsrv c:\test123.gho TestSession -t13:30
```

starts a Multicast session called TestSession and uses the file

c:\test123.gho. The connecting client's IP address will be displayed on screen. At half past one in the afternoon (1:30pm) the session transmission is started automatically.

Example 6:
Time-based and client count auto-start

```
ghostsrv c:\test123.gho TestSession -t13:30 -n10
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. At either half past one in the afternoon (1:30pm), or after 10 clients join the session, transmission is started automatically. Note that **ghostsrv** does not wait for *both* conditions to be met.

Example7:
Time-based and client count auto-start and automatic closing

```
ghostsrv c:\test123.gho TestSession -t13:30 -n10 -c
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. At either half past one in the afternoon (1:30pm), or after 10 clients join the session, transmission is started automatically. **Ghostsrv** does not wait for *both* conditions to be met. Once the Multicast session is completed, **ghostsrv** is closed down as requested.

Example 8:
Problem solving

```
ghostsrv -la -ferrlog.txt -n10 c:\test123.gho
TestSession
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. Once 10 clients have connected, the session transmis-

sions is started automatically. A log file **errlog.txt** will be created for debugging purposes. Please note that using a log file will reduce the performance of the Multicast transmission.

Deploying the DOS Multicast Server dosghsrv.exe

The DOS *Ghost* Multicast server:

❖ uses identical files to the DOS application diskette as described in the setup instructions in Chapter 4 "Setting up DOS-based Multicast Client and Server", *except* the file **ghost.exe** is removed and is replaced by **dosghserv.exe**.
❖ uses the Packet Driver interface
❖ provides a command line user interface
❖ offers a DOS command line alternative to the Windows based *Ghost* Multicast Server
❖ TCP/IP configured through **wattcp.cfg**.

Dosghsrv.exe uses the same packet driver setup as the *Ghost* Multicasting client. Refer to the Packet Driver Setup for more details on setup requirements.
The command line syntax is:

```
dosghsrv filename session [-l[A|l|W|S|E]][-
nclient_count][-tstart_time][-d][-p]
```

where:

filename =   path and filename of disk image file to be distributed
session =   session name to use

and optionally:

-l   =   creates log file **rmllog.txt** with the log level specified
               (E,S,W,L or A)

-n   =   starts sending to session automatically after the specified
               client_count clients have joined the session

-t   =   starts sending to session automatically after the specified
               24-hour time occurs

-d   =   use dump from client mode (load to client is default)

-p   =   specifies partition mode operation. If loading to clients,
         the partition number must be given. If dumping from
         client no partition number is required to be specified.

Example 1:

Uploading a complete disk Image File from a client *Ghost* machine:

```
dosghsrv c:\test123.gho TestSession -d
```

Starts a Multicast session called TestSession and creates or overwites the file **c:\test123.gho**. The first connecting client's IP address will be displayed on screen, and the session will start automatically. The client PC indicates the drive to use for the image creation.

Example 2:

Uploading partitions from a client *Ghost* machine to an image file:

```
dosghsrv c:\test123.gho TestSession -d -p
```

Starts a Multicast session called TestSession and creates or overwites the file **c:\test123.gho**. The first connecting client's IP address will be displayed on screen, and the session will start automatically. The Client PC indicates the drive and partitions to be included in the image created.

Example 3:

General use - Multicasting a Disk image file

```
dosghsrv.exe c:\test123.gho TestSession
```

starts a Multicast session called TestSession and uses the file

c:\test123.gho. The connecting client's IP address will be displayed on screen.   To start the session transmission press any key when all clients have connected.

Example 4:
General Partition Use:

```
dosghsrv c:\test123.gho TestSession -p2
```

Starts a Multicast session called TestSession and uses second partition in the file **c:\test123.gho**. The connecting clients IP address will be displayed on screen. Once all Clients have connected, pressing any key starts the session transmission.

Example 5:
Client count auto-start:

```
dosghsrv.exe -n10 c:\test123.gho TestSession
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. The session transmission is started automatically when 10 clients have connected.

Example 6:
Time based auto-start:

```
dosghsrv c:\test123.gho TestSession -t13:30
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. The connecting client's IP address will be displayed on screen. At half past one in the afternoon (1:30pm) the session transmission is started automatically.

Example 7:
Time based and client count auto-start:

```
dosghsrv c:\test123.gho TestSession -t13:30 -n10
```

starts a Multicast session called TestSession and uses the file **c:\test123.gho**. At either half past one in the afternoon (1:30pm), or after 10 clients join the session, transmission is started automatically. **Dosghsrv** does not wait for *both* conditions to be met.

Example 8:
Problem solving:

```
dosghsrv.exe -la -n10 c:\test123.gho TestSession
```

starts a Multicast session called TestSession and uses the file c:\test123.gho. The connecting client's IP address will be displayed on screen. The session transmission is started automatically when 10 clients have connected. A log file **rmllog.txt** will be created for debugging purposes. Note that using a log file will reduce the performance of the Multicast session.

Deploying the *Ghost* Multicast Client

When using *Ghost* Multicasting the *Ghost* client application **ghost.exe** receives a Multicast copy of an image file by joining and listening to a server's session. Alternatively, the *Ghost* client can upload an image file to the Multicast server.

The *Ghost* application runs under DOS and uses a packet driver interface to the network card. The TCP/IP settings are stored in a configuration file titled **wattcp.cfg** that should be located in the same directory as **ghost.exe**.

As with all *Ghost* applications, DHCP, BOOTP and manual setting of IP addresses are supported.   See Appendix D for more information of the **wattcp.cfg** file and IP address assignment. Refer to the Multicast section in the Setting Up chapter for details.

Ghost Multicasting

The *Ghost* Multicast command line switches are listed in Appendix A. They include:

- -CLONE
- -JS
- -JL

the selection of the partition or drive to be written or read on the *Ghost* client for the multicasting session is specified on the *Ghost* client. Follow online prompts, or refer to the appropriate section in the *Ghost* Procedures in Chapter 6 .

Chapter 6

Ghost Procedures

Overview

This chapter describes how to perform *Ghost* operations and procedures.   The procedures in this chapter assume that the setting up of the hardware, as described in Chapter 4, "Setting Up, has been completed.

To run *Ghost,* close down Windows and, from the DOS prompt, type **ghost.exe**. Alternatively, boot the machine using the boot disk created in Chapter 4, "Setting Up". The *Ghost* menu screen is shown below.

Managing Image Files

Ghost includes the capability to create a file that contains all the information required to recreate a complete disk or partition. This file is known as the *Ghost image file* and has a .**gho** extension. Image files are a useful way of being able to store and reliably compress images of model system configurations, or create backup copies of complete drives or partitions.

The image files created with *Ghost* 5.0 not only contain the entire disk and/or partitions in the disk, but also offer:

❖ various levels of compression:
❖ CRC32 data integrity checking
❖ splitting of media files and
❖ spanning across volumes.

With the newly introduced Windows *Ghost-Explorer* application, a *Ghost* image file companion utility, individual files from these image files can be recovered and restored with out the need for the complete partition or disk to be restored.

Image Files and Compression

Image files created in *Ghost* 5.0 support several levels of data compression. When using *Ghost* in interactive mode, three compression options are provided as shown in the example below.
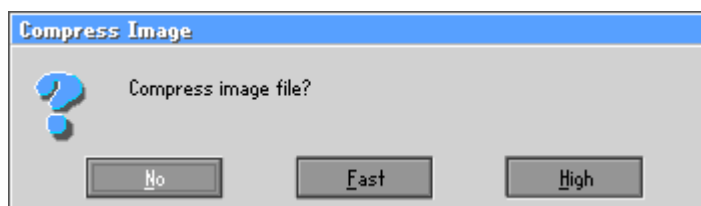
The *Ghost* switches provide access to nine levels of compression. The Compression switch -Z is detailed in Appendix A.

As a general rule, the more compression you use, the slower *Ghost* will operate. There is a big difference in speed between high compression and no compression when creating an image file on a local disk. Over a network or NetBIOS connection, fast compression is often as fast as, or faster than, no compression. Over a parallel cable, high compression is often faster than no compression because fewer bytes need to be sent over the cable. Decompression of high-compressed images is much faster than the original compression. The level of compression you should select depends on your own individual requirements.

Image Files and CRC32

When image files are created, CRC32 details are embedded into the file to ensure image file corruption is detected when it is being restored to disk. CRC32 is currently included on a file-by-file basis

with FAT partitions and on a MFT Table basis for NTFS partitions. Image All Partitions are not CRC Checked.

In addition to image file error detection, the CRC values can be used to verify that image files and partitions or disks are identical. This can offer an additional detection method against bad sector writes and other drive anomalies that may be missed during normal imaging checks.

A text file containing CRC values and associated file attributes can be generated using the -CRC32 *Ghost* command line switch. These switches and functions are detailed in Appendix A.

Image Files and Volume Spanning

Standard Image Files

Standard image files consist of a single file that contains the contents of the complete disk, or required partitions. This type of image file is usually used for storing system configurations on server network drives for later restoration, or on other hard drives, and tape drives where the volume is large enough and capable of holding the complete image file in one piece.

Size-limited multi-segment image files

There are situations where it may not be practical to have a standard image file. *Ghost* offers the ability to split up an image file into segments (known as 'spans') that are limited to a size specified by the user. For example, you may wish to keep files created on your network drive limited to 100MB in size so you can transfer them more easily in the future. This option is most commonly used to limit span sizes to 550Mb for later transfer onto CD-ROM.

Spanned image files

Spanned image files are similar to size-limited multi-segment image files. The difference is that each segment file (or 'span') of the image file is limited by the actual volume size of the media the image is being saved to. This allows the user to specify a drive and filename, and let

Ghost sort out when to request another volume or location for the remaining data. For example, this is very useful when using ZIP, JAZ, LS120 SuperDisk and other disk drive types.

Ghost also allows size limiting of spans when spanning volumes, ensuring no span exceeds the maximum size.

With all image files, the only constraint on the selection of the destination volume is that it must not be part of the source selection, e.g, it cannot be on a source disk or partition if that disk or partition is being included in the image.

Creating an image file

An image file can be created using the Disk-to-Image file and Partitions-to-Image file options in *Ghost.* Refer to the Procedures later in this section for more information.

### How to span an image across multiple volumes and limit span sizes

When performing a 'disk-to-image file', or a 'partition to image file', if the destination drive does not have sufficient free space to accommodate the image, the user is first informed that there is not enough space on the destination volume, and is asked whether the image file is to be compressed. *Ghost* assumes compression will reduce the size of the image by 1/3 (one-third) when determining whether the image will fit. (Alternatively, the -span and -split *Ghost* command line switches can be used to configure *Ghost* on start up to use image file splitting. These switches are detailed in Appendix A.)

The following message displays:

Compress Image

Not enough space - use compression?

No   Fast   High

If the user selects compression and there is still a possibility of insufficient space, the following message displays:

Question:

Do you wish to enable spanning?

No   Yes
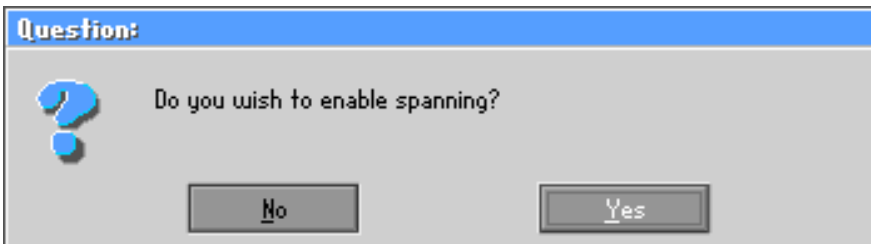
If Spanning is not enabled, an error message displays:
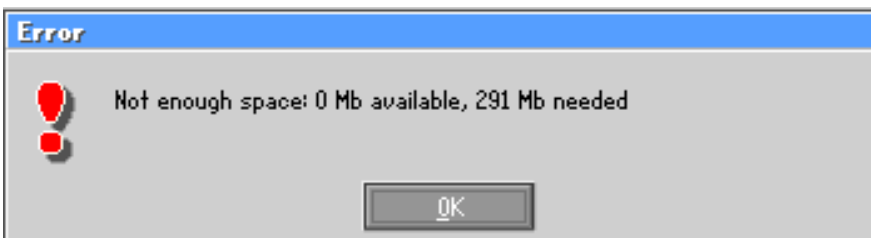
Error

Not enough space: 0 Mb available, 291 Mb needed

OK

Before starting to dump the disk contents to the image file, *Ghost* displays the source and destination details and gives you a chance to back out. The default is to back out.

Once the process starts, the dump continues until the destination volume is filled up. The user is then prompted to either select <Enter> to continue, or specify where the next 'span' of the image file is to be located.

Either select <OK> to continue on the same form of media or enter a filename to span to a different location.
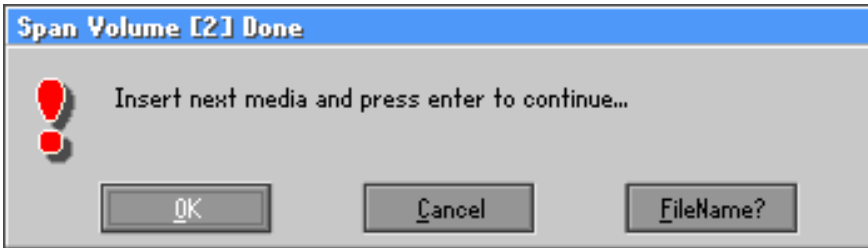
Example:

If you started spanning onto a JAZ drive and wish to span a 3.0 Gigabyte drive onto just JAZ disks, just select <Enter> to continue on JAZ disks. If you wish to span across different forms of media, selecting "Filename?" gives you the option to span onto a different location.

Caution: Record where you save your segments of the 'span'. Also record what filename each 'span' segment is. Ghost will not record where and what filename you have selected.

If you have a single partition on a drive, or if you are imaging a single partition, *Ghost* will end on the last-spanned volume with no user intervention. However, if you are imaging a hard drive with multiple partitions, *Ghost* needs to record boundary information onto the first span of the image file. This boundary information is recorded to the location of the partition amongst the spanned set. The user is prompted to confirm that the first span is ready to be updated.

The screen below shows how *Ghost* will prompt you for span set disk one and for subsequent volumes.
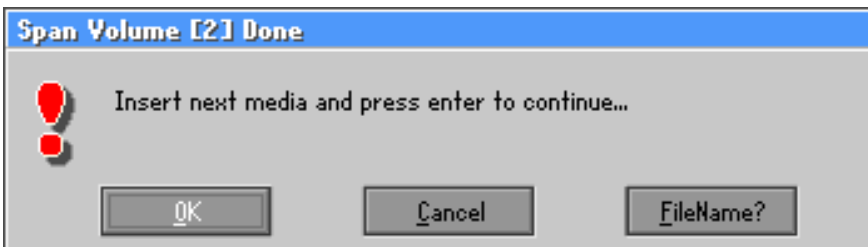
```
Span Volume [2] Done
```

Insert next media and press enter to continue...

[OK]   [Cancel]   [FileName?]

When the partition dump is complete, the user is returned to the menu screen.

How to load from a spanned image

When loading a disk or partition from an image file, the process is the same as loading from an unspanned image file. The loading procedure is the reverse of the dumping procedure. The user is prompted to provide details of each portion of the spanned image, as shown in the screen below.

```
Span Volume [2] Done
```

Insert next media and press enter to continue...

[OK]   [Cancel]   [FileName?]

Either:

❖ select <OK> to continue on the same form of media. Example - If you originally spanned onto a JAZ drive and wish to restore a 3.0 Gigabyte drive from just JAZ disks, replace the disk and just hit <Enter> to continue from JAZ disks.

or,

❖ if you wish to restore from different forms of media, selecting "**Filename?**" gives you the option to restore from a different location. **Caution**: You need to know where you saved your segments of the 'span'. You must also know each filename and path, for each 'span' segment.

When the disk image load is complete, the target PC will need to be rebooted. By default *Ghost* will display the following screen.
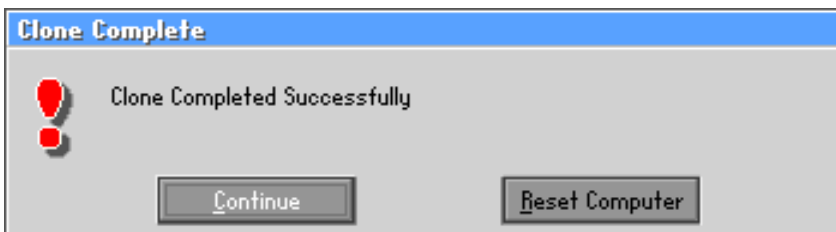
Image Files and Tape Drives

Ghost's support of SCSI Tape drives allows it to store a single image file onto a tape. When written onto the tape, there is no associated file system used and this means that you are unable to access the tape from a drive letter as if it were another storage drive. SCSI Tapes only support Standard Image files.

When using tape drives with *Ghost*, the tape drive can be selected as the source or destination device in the File Locator Window. Each SCSI tape device is shown as MTx where x is a number starting at 0 and incrementing for each drive present.

For example, the following shows a tape drive MT0 available for use in the File Locator window.
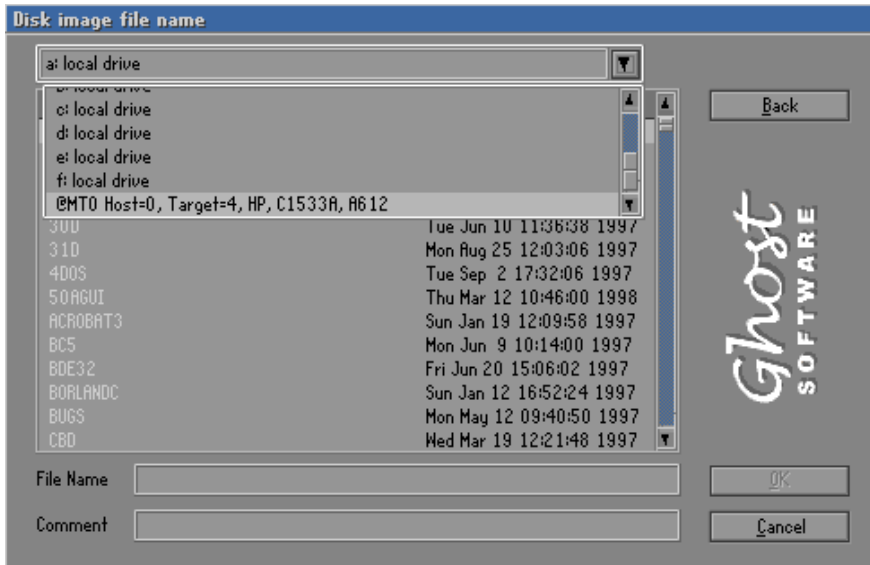
For *Ghost* to be able to access SCSI tape drives, a DOS ASPI driver must be installed prior to use. Further details are available in Chapter 4 - Setting Up.

Ghost in its default mode performs well with most SCSI tape devices. In some situations with older style SCSI tape devices and possibly with unreliable tapes, *Ghost* may need to be configured to slow down, or alter the way it uses the tape device. These options are listed in Appendix A.

Image files - Content Viewing/Selective file Restoration

Ghost-Explorer is a *Ghost* companion Windows95 program that looks similar to Windows Explorer. It can open a *Ghost* image file and restore individual files, or entire directory structures from it. It can also launch *Ghost* in batch mode to create an image file, using Windows dialog boxes to prompt for the disk and partition to dump. A sample screen is shown below:
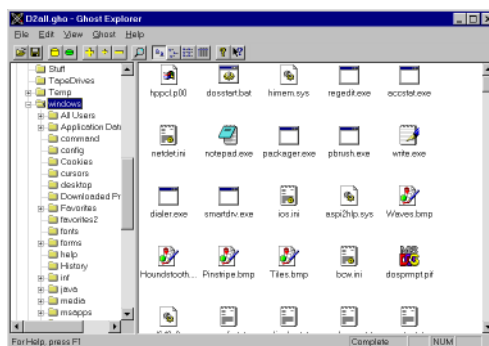
Ghost Version 4.0a, and beyond, creates an index in the image file as it works that allows *Ghost-Explorer* to create a display of files very quickly. Image files created by earlier versions of *Ghost* can still be viewed with *Ghost-Explorer*, but there is a possibly lengthy delay while *Ghost-Explorer* reads through the image file to build an index.

Ghost and Bad Sectors on FAT File Systems

Ghost includes the ability to handle bad clusters on FAT partitions when it detects them or if additional checks are performed. *Ghost* will only become aware of bad sectors when its read/write attempt on the hard drive fails through normal usage. *Ghost* offers the following switches:

❖ verification of disk writes (-vdw)
❖ handling of bad FAT clusters (-bfc)

Ghost in default mode will stop and report any noted bad clusters. To improve upon this, the verification switch can be used to provide a higher level of certainty that no clusters located in the file space of a FAT partition are bad. As well as this, the bad cluster handling switch allows *Ghost* to keep track of bad clusters in FAT partitions. Like other FAT disk utilities, *Ghost* is unable to handle the case of bad clusters in a FAT table.

In addition, CRC32 checking can be used to ensure data integrity.

Ghost Operations:

This section describes *Ghost* operations. *Ghost* has the ability to:

❖ clone drives/partitions
❖ create image files
❖ restore image files
❖ perform all of the above using various connection techniques.

Ghost copies every required partition, regardless of type, from the source (disk or disk-image file) to the destination. If the source and destination disks were identical in size and structure then all that would be needed is a sector by sector copy; but in practice this is seldom the case. *Ghost* positions each partition or logical drive on the target disk using the same rules as FDISK. In addition, if the partition is a FAT12, FAT16,

FAT32, or NTFS type (which most PC disks are), *Ghost* allows the target partition to be expanded or contracted to suit.

Ghost Main Menu Options:

From the *Ghost* menu the options are as follows. Note that unavailable options are greyed out. This may be because the license does not cover the option, or the hardware is not set up to support the option.

Local
Used for operations involving:

❖ Internal disk drives only
❖ locally attached SCSI tape devices
❖ locally attached removable media devices
❖ locally mapped file server volumes access.

LPT
Used for operations between two peer computers connected with a laplink style cable.

Multicasting
Used for image file uploading and downloading from the *Ghost* Multicast Server application.

NetBIOS
Used for operations between two peer computers connected with Network Interface Cards and cable.

LOCAL Action Options:
❖ Disk - procedures involving complete disk cloning.
❖ Partition - procedures involving individual partition cloning.
❖ Check - procedures involving checking of disks and images.

LPT and NetBIOS Master Action Options:
❖ Disk - procedures involving complete disk cloning.
❖ Partition - procedures involving individual partition cloning.

- ❖ Check - procedures involving checking of disks and images.
- ❖ Back - allows navigation to previous screen.

Master and Slave
- ❖ The term **Master** indicates the machine from which you control the connection.
- ❖ The term **Slave** indicates the other machine(s) participating in the connection

Refer to the table for details on selecting Slave and Master in the Setting Up chapter.

LPT and NetBIOS Slave Action Options:
None - determined by the connected Master.

Multicasting Options:

After establishing a session on the *Ghost* multicast server, the action is determined by the *Ghost* multicast server.

Disk Options:

To disk - Select this option if you wish to copy all disk contents including all partitions from one disk to another.

To image file - Select this option if you wish to save an exact image copy of a disk into an image file. The file can be on a local disk, a server drive, or on the drive of a PC connected via NetBIOS or the LPT port.

From image file - This is the reverse of the previous option. Select this if you wish to load the contents of a disk image file onto the disk of the local PC. The image file must have been previously created by *Ghost.*

Partition Options

To partition - Select this option if you wish to copy a disk partition to another partition. The destination partition can be on a local disk or on the disk of a PC connected via NetBIOS or the LPT port.

To image file - Select this option if you wish to save an exact image copy of one or more disk partitions in a file. The file can be on a local disk, a server drive, or on the drive of a PC connected via NetBIOS or the LPT port.

From image file - This is the reverse of the previous option. Select this if you wish to load the contents of a partition image file onto a partition of the local PC. Note that although the file can contain an image of more than one partition, only one partition can be selected for loading from the file. The file can be on a local disk, a server drive, or on the drive of a PC connected via NetBIOS or the LPT port. The image file must have been previously created by *Ghost.*

Check Options:

Image file - Select this option if you wish to check the integrity of an image file.

Disk - Select this option if you wish to check the file structure of a disk.

Options:

When you select 'Options' from the main menu, an Options screen displays. Selecting one of the options on this screen will direct you to the appropriate section of the online help. Refer also to Appendix A for switch references.
.

Ghost Step-by-Step Procedures

Disk Options

The following procedures are accessed from the *Ghost* main menu by selecting either:

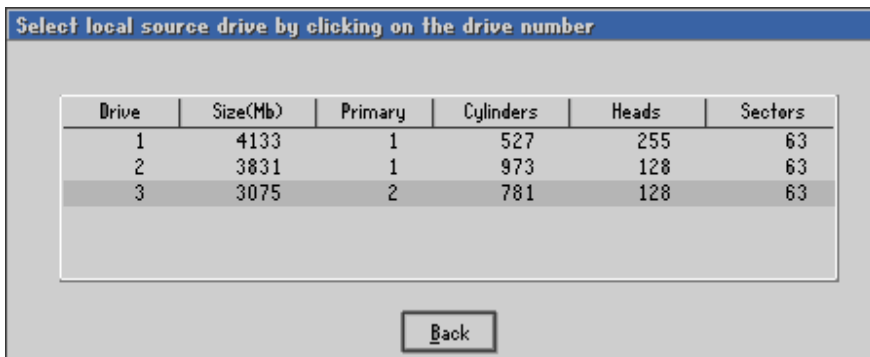❖ Local
❖ LPT | Master
❖ NetBIOS | Master.

Disk-to-disk procedure

There are four steps as follows:
1. Select source drive.
2. Select destination drive.
3. Confirm destination drive details.
4. Proceed with disk clone.

Step 1 - Select source drive

From the *Ghost* menu select **Disk|ToDisk**. From the dialogue box select the source drive.

Select local source drive by clicking on the drive number

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	1	973	128	63
3	3075	2	781	128	63

Back

Ghost displays the details of every disk it can find on the local PC and allow you to select the disk you wish to copy <u>from</u>.

Step 2 - Select destination drive
From the dialogue box, select the destination drive.

Select local destination drive by clicking on the drive number					
Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	1	973	128	63
3	3075	2	781	128	63

Back
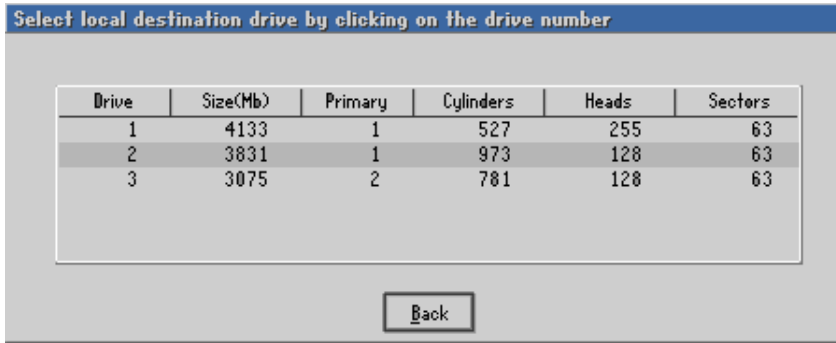
Ghost displays the details of every disk it can find on the destination PC and allow you to select the disk you wish to copy to. If a peer-to-peer connection is established this will be the slave PC's drives. (If this is a local disk-to-disk copy, then the source disk will be unavailable for selection.) ***Warning****: Choose carefully as this is the disk that is going to get overwritten!*

Step 3 - Confirm destination drive details

Ghost displays a suggested partition layout for the target disk.

Destination Drive Details

| Part	Type	Description	Label	New Size	Old Size	Data Size
1	07	NTFS	WinNT	2327	1500	221
2	06	Fat16 extd	NO NAME	752	752	86
3	0b	Fat32 extd	NO NAME	752	752	14
			Free	0	70	
			Total	3831	3075	321

Back   OK

By default *Ghost* allocates any extra space that the new disk has to the first FAT or NTFS partition that it discovers.

You can change the size of any target FAT or NTFS partition at this stage, simply by entering the new size (in megabytes).

Ghost will not allow you to enter a value that exceeds the available space, or that is not large enough to contain the data held in the source partition. When you are satisfied with all details select <OK> to continue.

Step 4 - Proceed with disk clone

You will be prompted to proceed with the disk cloning. *Ghost* displays the source and destination details and this is your chance to back out.
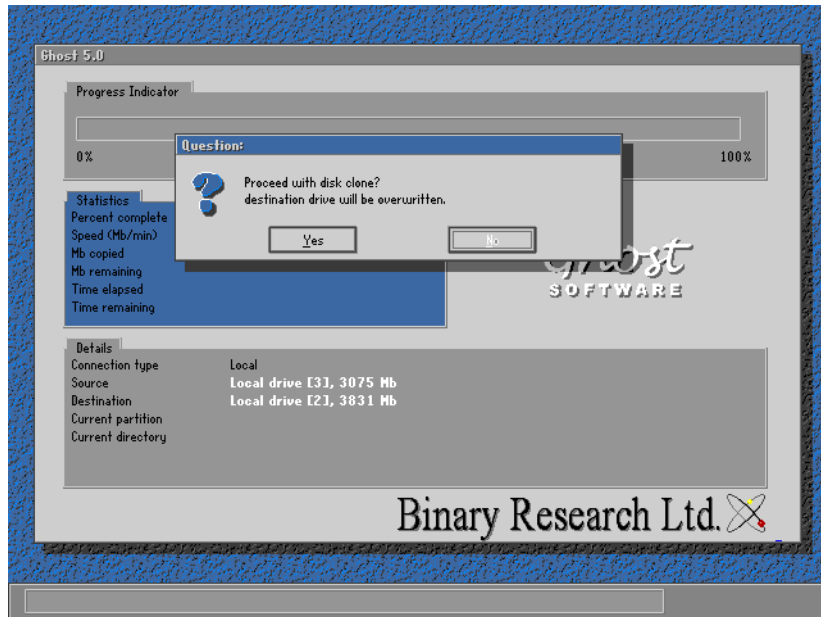
Look at the details displayed and <u>ensure you have chosen the correct options</u>. ***Caution****: Only select 'Yes' if you are really sure*, as the destination drive will be completely overwritten with NO chance of recovering any data.

Select 'Cancel' to return to the menu, or Select 'Yes' to proceed. The system performs a quick integrity check on the file structure and then copies the files to the new drive. If you need to abort the process use <Ctrl-C>, but be aware that this leaves the destination drive in an unknown state.

When the process is complete you may wish to check the integrity of the destination drive.
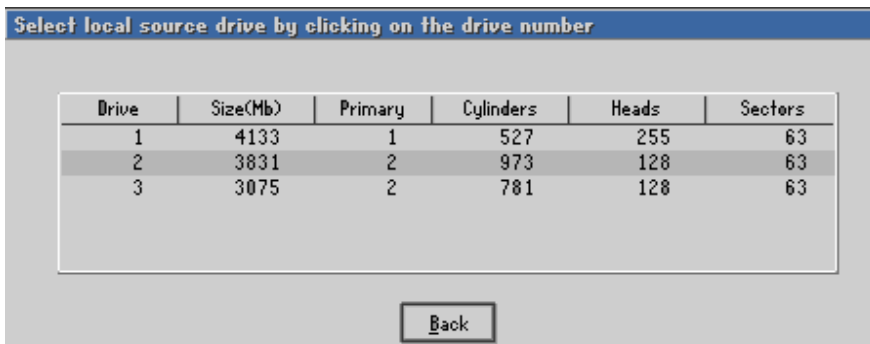
Disk to image file procedure

There are three steps as follows:

1. Select the source drive.
2. Enter the filename to dump the disk image to.
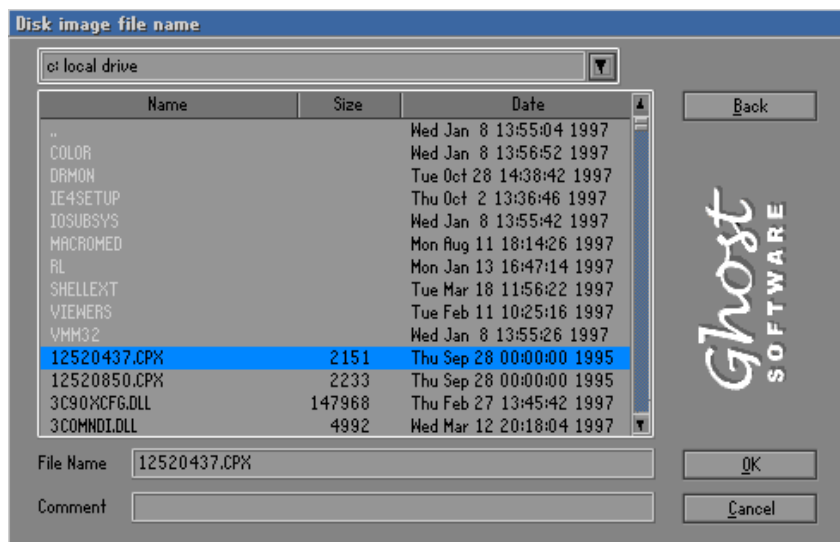3. Proceed with disk copy.

Step 1 - Select the source drive

From the *Ghost* menu select **Disk|To Image**. From the dialogue box select the source drive. *Ghost* display the details of every disk it can find on the local PC and allow you to select the disk you wish to copy from.

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	2	973	128	63
3	3075	2	781	128	63

Select local source drive by clicking on the drive number

Back

### Step 2 - Enter the filename to dump the disk image to

In the File Locator box enter (or browse for) the image file. Specify the drive or device and select the full path name.

Note that the image file may reside on either a locally mapped net-work fileserver volume (the most common option), or a local drive (but not the one that is being copied from). When using peer-to-peer connections, the file will be located on the slave *Ghost* machine.
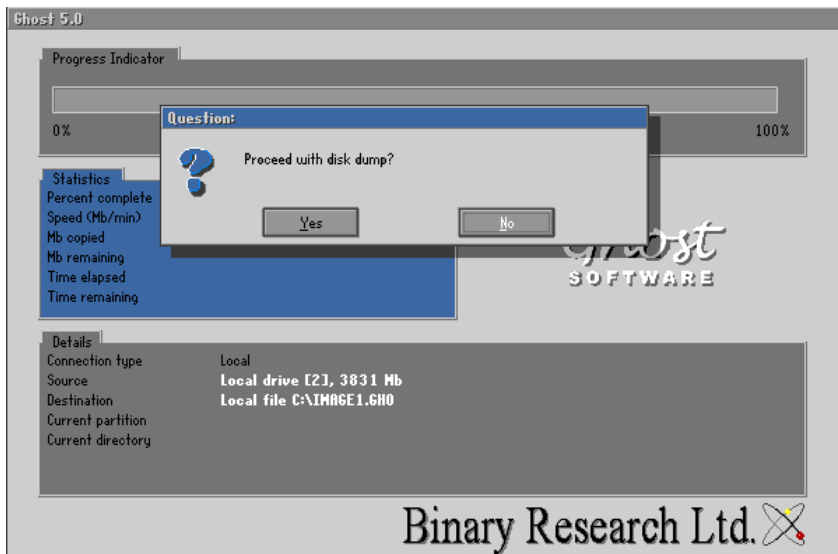
Step 3 - Proceed with disk copy

Compression options display:
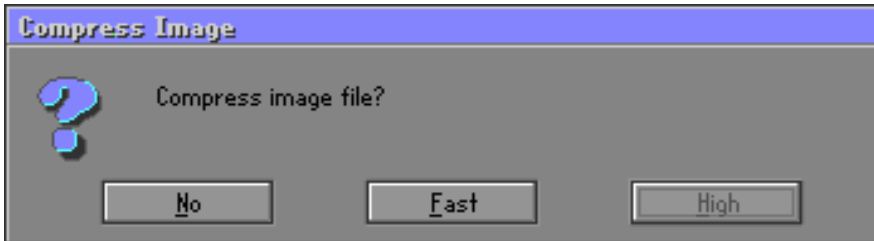
Compress Image

Compress image file?

| No | Fast | High |

- ❖ Select 'No' for no compression (high speed)
- ❖ Select 'Fast' for low compression (medium speed)
- ❖ Select 'High' for high compression (slower speed)

When compression is used, copying operations are slower. For this reason, it is recommended that compression be used to reflect your individual usage requirements.

Before starting to dump the disk contents to the image file, *Ghost* displays the source and destination details and gives you a chance to back out. (The default is to back out.)

Ghost 5.0

Progress Indicator

0%   100%

Statistics
Percent complete
Speed (Mb/min)
Mb copied
Mb remaining
Time elapsed
Time remaining

Question:

Proceed with disk dump?

Yes   No

Details
Connection type   Local
Source   Local drive [2], 3831 Mb
Destination   Local file C:\IMAGE1.GHO
Current partition
Current directory

SOFTWARE

Binary Research Ltd.

The system checks the directory structure and then proceeds to dump

the files to the selected image file. A progress indicator displays the status of the process.
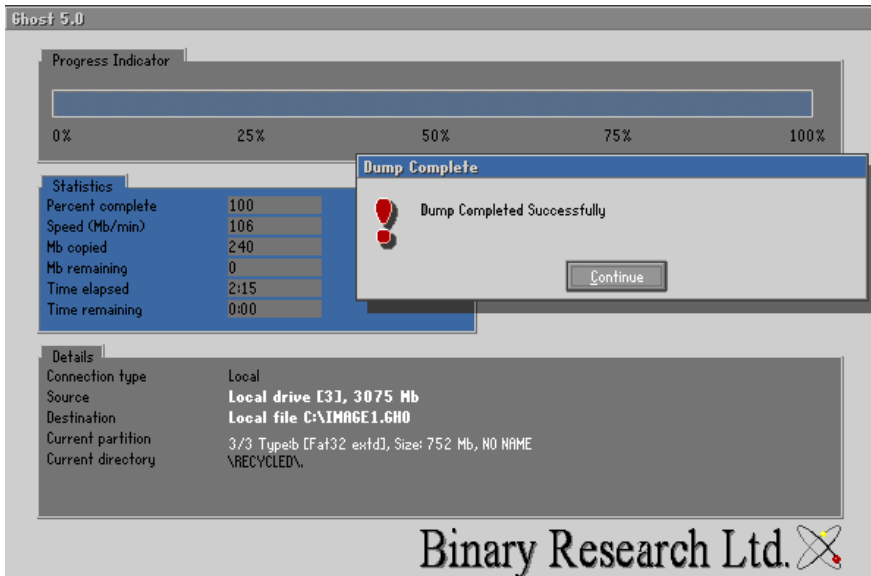
If spanning is enabled, *Ghost* will prompt for additional disks and volumes required as detailed in the Image file management section. When the process has completed the user is returned to the *Ghost* main menu.

Disk from image file procedure

There are four steps as follows:

1. Enter the filename to load the disk image from.
2. Select the destination drive.
3. Confirm the destination drive details.
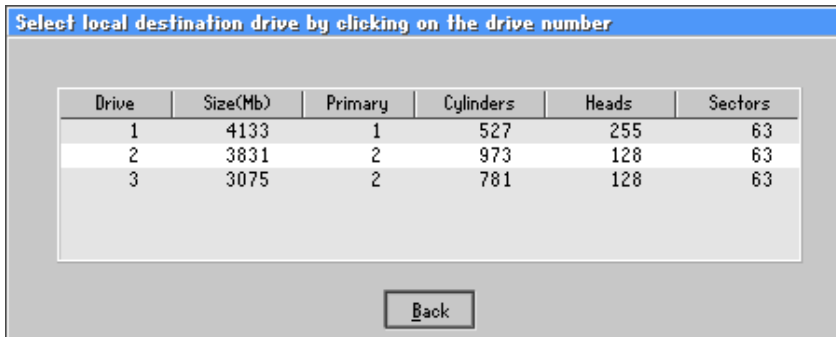4. Proceed with the disk load.

110

Step 1 - Enter the filename to load the disk image from

In the File Locator box enter (or browse for) the image file. Specify the drive or device and select the full path name.

Note that the image file may reside on either a locally mapped network fileserver volume (the most common option), or a local drive (but not the one that is being copied from). When using peer-to-peer connections, the file will be located on the slave *Ghost* machine.

Step 2 - Select the destination drive

Ghost displays the details of every disk it can find on the local PC and allows you to select the disk you wish to copy <u>to</u>.
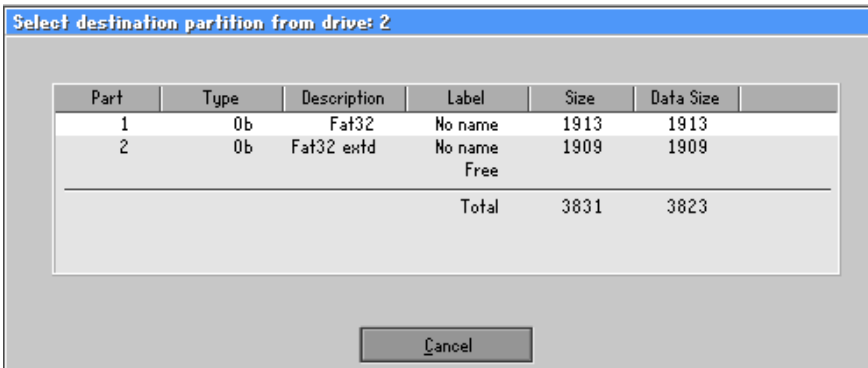
Select local destination drive by clicking on the drive number

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	2	973	128	63
3	3075	2	781	128	63

[<u>B</u>ack]

The source disk will be unavailable for selection. ***Warning****: Choose carefully as this is the disk that is going to get overwritten!*

111

Step 3 - Confirm destination drive details

Ghost displays a suggested partition layout for the destination disk. By default *Ghost* allocates any extra space that the new disk has to the first FAT or NTFS partition that it discovers.

Select destination partition from drive: 2

| Part	Type	Description	Label	Size	Data Size	
1	0b	Fat32	No name	1913	1913	
2	0b	Fat32 extd	No name	1909	1909	
			Free			
			Total	3831	3823	

Cancel

You can change the size of any target FAT or NTFS partition at this stage, simply by entering the new size (in megabytes).

Ghost will not allow you to enter a value that exceeds the available space, or that is not large enough to contain the data held in the source partition. When you are satisfied with the allocation of space between partitions, select <OK>.

Step 4 - Proceed with disk load

Before *Ghost* starts to overwrite the destination disk, the source and destination details are displayed and you have a chance to back out. The default is to back out.
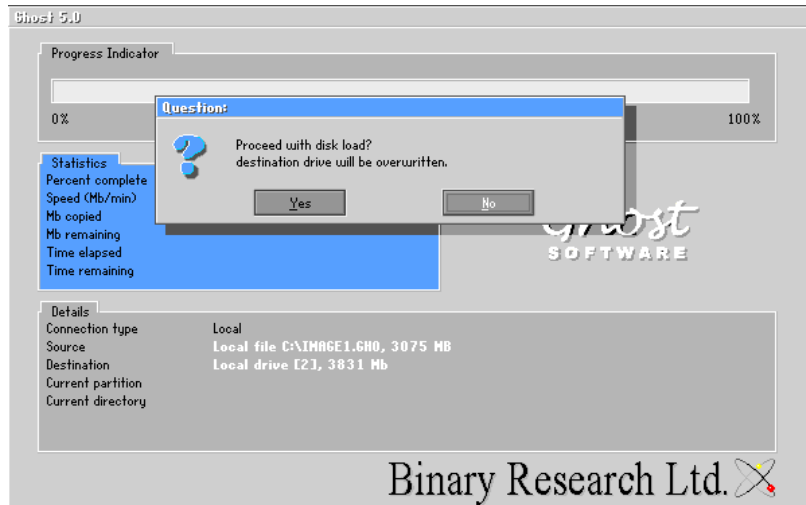
Warning*: Only select YES if you are really sure*. The target drive will be completely overwritten, with NO chance of recovering any data.

Once the copy has started it is possible to abort the process by pressing <Ctrl-C>, however the destination disk will be left in an unknown state.

Spanned and split image files are handled as outlined in the Image File Management section.

When the disk image load is complete you will need to reboot the machine.
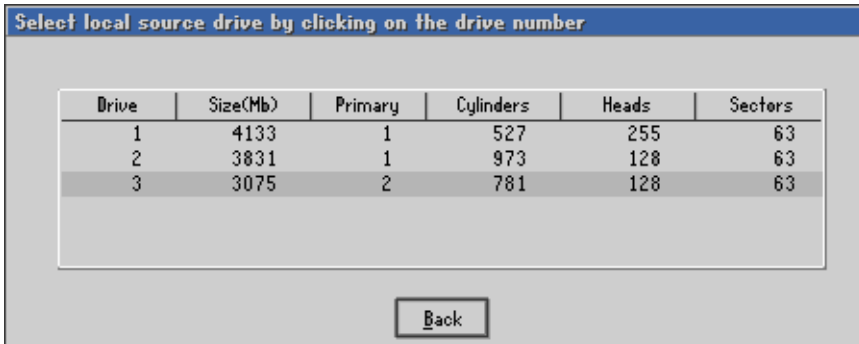
Partition-to-partition procedure

There are five steps as follows:

1. Select source drive
2. Select source partition
3. Select destination drive
4. Select destination partition
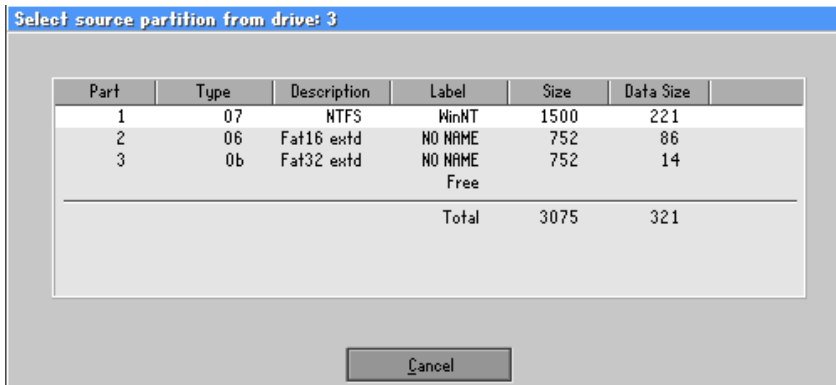5. Proceed with partition clone.

Step 1 - Select the source drive

Ghost displays the details of every disk it can find on the local PC and allow you to select the disk you wish to copy <u>from</u>.

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	1	973	128	63
3	3075	2	781	128	63

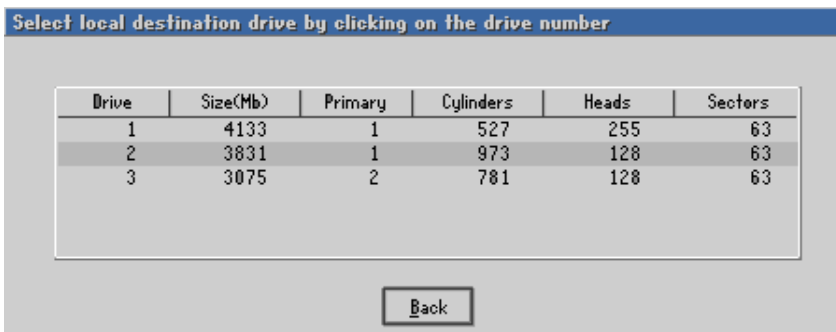Select local source drive by clicking on the drive number

Back

Step 2 - Select the source partition

Ghost displays the details of all partitions it can find on the selected source drive and allow you to select the partition you wish to copy <u>from</u>.

Part	Type	Description	Label	Size	Data Size	
1	07	NTFS	WinNT	1500	221	
2	06	Fat16 extd	NO NAME	752	86	
3	0b	Fat32 extd	NO NAME	752	14	
			Free			
			Total	3075	321	

Select source partition from drive: 3

Cancel

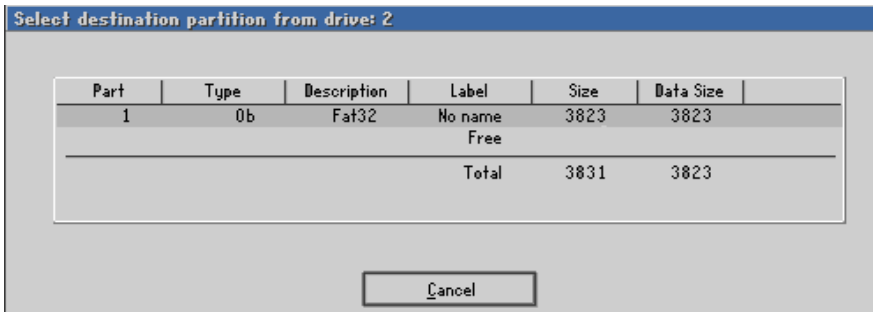### Step 3 - Select the destination drive

Ghost displays the details of every disk it can find on the destination PC and allows you to select the disk you wish to copy the partition <u>to</u>. For peer-to-peer connections, the *Slave Ghost* machine is the destination.

Select local destination drive by clicking on the drive number

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	1	973	128	63
3	3075	2	781	128	63

Back

115

Step 4 - Select the destination partition

Ghost displays the details of all the partitions it can find on the se-
lected destination drive and allows you to select the partition you wish
to copy <u>to</u>. **Warning**: *Choose carefully as this is the partition that is
going to be overwritten!*

Select destination partition from drive: 2						
Part	Type	Description	Label	Size	Data Size	
1	0b	Fat32	No name	3823	3823	
			Free			
			Total	3831	3823	

Cancel

Step 5 - Proceed with partition copy

Before starting to overwrite the destination partition, *Ghost* displays
the source and destination details and gives you a chance to back out.
The default is to back out.

Warning: *Only select 'Yes' if you are really sure*. The target partition will be completely overwritten, with NO chance of recovering any data.

Once the copy has started it is possible to abort the process by pressing <Ctrl-C>, however the destination partition will be left in an unknown state.

When the partition copy is complete, the destination PC will need to be rebooted.

Partition-to-image file procedure

There are four steps as follows:

1. Select the source drive
2. Select the source partition(s)
3. Enter the filename to copy the partition image to
4. Proceed with partition dump.

Step 1 - Select the source drive

Ghost displays the details of every disk it can find on the local PC and allows you to select the disk you wish to copy <u>from</u>.

Step 2 - Select the source partition(s)

Ghost displays the details of all the partitions it can find on the selected source drive and allows you to select the partition(s) you wish to include in the partition image file. Highlight partitions you wish to include.

Select source partition(s) from drive: 3

Part	Type	Description	Volume Label	Size in Mb	Data Size in Mb	
1	07	NTFS	WinNT	1500	221	
2	06	Fat16 extd	NO NAME	752	86	
3	0b	Fat32 extd	NO NAME	752	14	
			Free	70		
			Total	3075	321	

[OK]   [Cancel]

Step 3 - Enter the filename to copy the partition image to

In the File Locator box enter (or browse for) the image file. Specify the drive or device and select the full path name.

Note that the image file may reside on either a locally mapped network

fileserver volume (the most common option), or a local drive (but not the one that is being copied from). When using peer-to-peer connections, the file will be located on the slave *Ghost* machine.

Step 4 - Proceed with partition dump
Compression options display:

- ❖ Select 'No' for no compression (high speed)
- ❖ Select 'Fast' for low compression (medium speed)
- ❖ Select 'High' for high compression (slower speed).

When compression is used, copying operations may be slower. For this reason, it is recommended that compression be used to reflect your individual usage requirements.

Before starting to dump the partition contents to the image file, *Ghost* displays the source and destination details and gives you a chance to back out. The default is to back out.

When the partition dump is complete, the user is returned to the *Ghost* main menu.

Partition-from-image file procedure

There are five steps as follows:

1. Enter the filename to load partition image from.
2. Select the source partition from the image file.
3. Select the destination drive.
4. Select the destination partition.
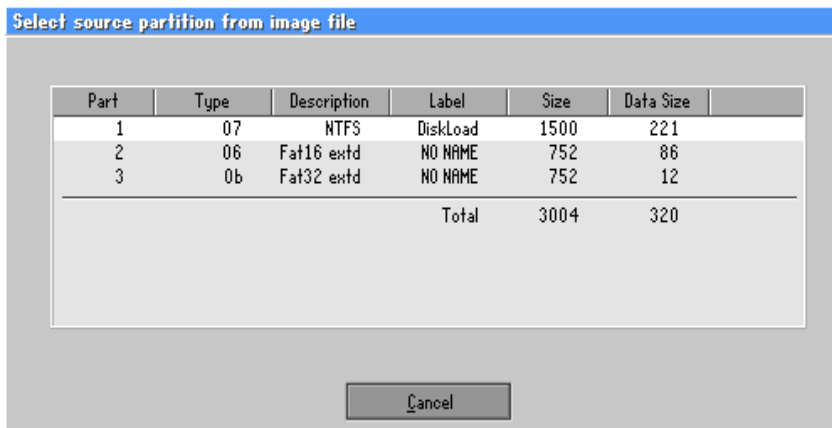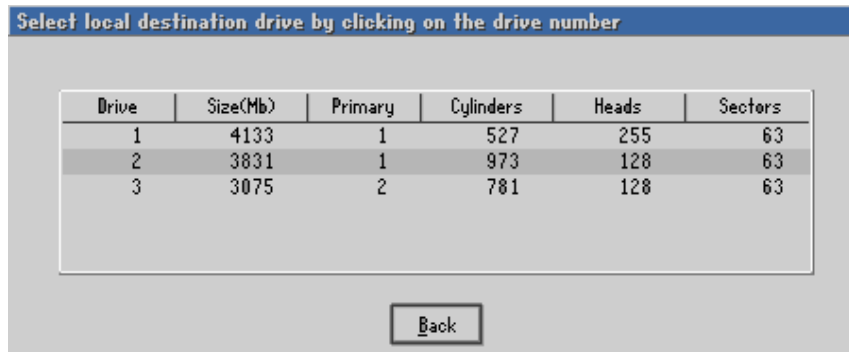5. Proceed with the partition load.

Step 1 - Enter filename to load partition image from

In the File Locator box enter (or browse for) the image file. Specify the drive or device and select the full path name.

Note that the image file may reside on either a locally mapped network fileserver volume (the most common option), or a local drive (but not the one that is being copied from). When using peer-to-peer connections, the file will be located on the slave *Ghost* machine.

Step 2 - Select the source partition from the image file

Ghost displays the details of the partitions it can find in the image file and allows you to select the partition you wish to copy <u>from</u>.

Select source partition from image file

Part	Type	Description	Label	Size	Data Size	
1	07	NTFS	DiskLoad	1500	221	
2	06	Fat16 extd	NO NAME	752	86	
3	0b	Fat32 extd	NO NAME	752	12	
			Total	3004	320	

Cancel

Step 3 - Select the destination drive

Ghost displays the details of every disk it can find on the local PC and allows you to select the disk you wish to copy the partition <u>to</u>.

Select local destination drive by clicking on the drive number

Drive	Size(Mb)	Primary	Cylinders	Heads	Sectors
1	4133	1	527	255	63
2	3831	1	973	128	63
3	3075	2	781	128	63

Back

Step 4 - Select the destination partition

Ghost displays the details of all the partitions it can find on the selected destination drive and allows you to select the partition you wish to write the image file partition <u>to</u>.
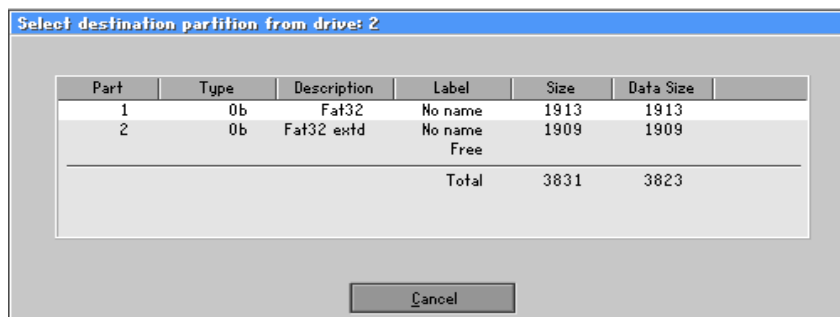
Select destination partition from drive: 2

Part	Type	Description	Label	Size	Data Size
1	0b	Fat32	No name	1913	1913
2	0b	Fat32 extd	No name	1909	1909
			Free		
			Total	3831	3823

Cancel

121

Warning: *Choose carefully as this is the partition that is going to be overwritten!*

Step 5 - Proceed with partition load

Before starting to overwrite the destination partition, *Ghost* displays the full source and destination details and gives you the chance to back out. The default is to back out.
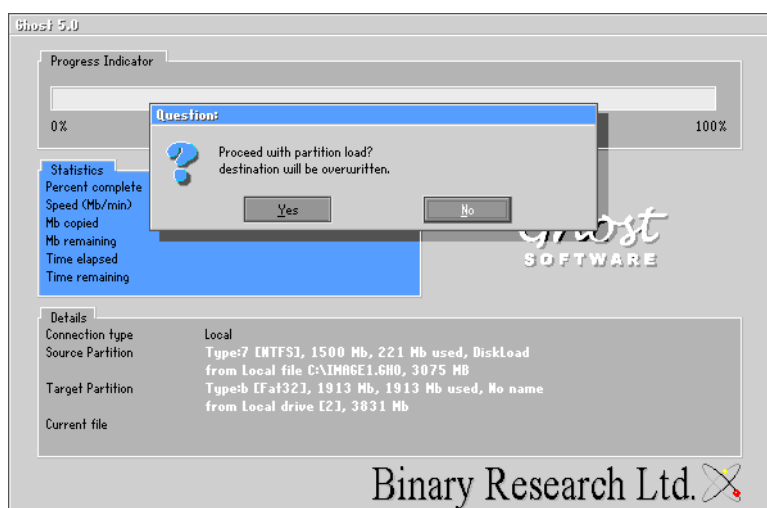
Warning: *Only select 'Yes' if you are really sure.* The target partition will be completely overwritten with NO chance of recovering any data.

Once the load has started it is possible to abort the process by pressing <Ctrl-C>, however the destination partition will be in an unknown state.

When the partition load is complete the target PC will need to be rebooted.

Note: It is important to remember that a destination drive's partition needs to exist before *Ghost* can load an image file to it.

Concepts

The use of *Ghost* in most situations to perform basic functions requires very little understanding of the computer system or hardware on which it operates. This chapter introduces the main concepts and hardware fundamentals that relate to *Ghost* and its usage.

Hard Disk Drive Concepts:

What is a hard disk?

A hard disk is used to store data and instructions in a computer system. Its basic physical components are:

❖ platters - rigid aluminum disks coated with magnetic particles
❖ spindle - the platters rotate on the spindle
❖ read/write heads - write and read the data from the platters.

What is a Bad Sector?

Data is recorded on hard disks as a series of bits, each bit being stored as either a positive or negative charge on the layer of iron oxide. Sometimes, following extensive use, or other abnormal operating conditions, the magnetic coating deteriorates, leading to difficulty when writing and reading to parts of the disk. These areas of deterioration are known as 'bad sectors'.

Storing and Retrieving Data

The platters rotate, all at the same speed, on the spindle while the read/write heads (both above and below each platter) move across and back over the surface of the platter.

When data is to be saved, the computer sends it as a series of bits to the hard disk. The hard disk then moves the arms backwards and forwards over the platter to allow the read/write heads to magnetically record (write the bits randomly onto the surface). Random recording allows faster retrieval than sequential recording, as used in tape systems.

Disk Formatting

Disk formatting determines how the bits of data are organized on the disk. The disk is formatted before data is recorded so that files (in bit form) can be efficiently written and retrieved.

Hard disks are formatted in two ways, at first physically, and then logically. The hard disk must be formatted physically before it can be formatted logically. This physical formatting is done by the drive manufacturer. *Ghost* logically formats at the same time as entering the data, thus saving time and improving speed.

Physical formatting

Physical formatting is also called low-level formatting, and is usually performed by the hard disk manufacturer. Its purpose is to divide the hard disk platters into elements called tracks, sectors, and cylinders, for recording and reading data methodically.

Tracks are numbered and appear on both sides of a platter. They are arranged as circular paths in the same way as those on musical compact disks. The numbering sequence for disks starts from the outer edge, while for CD-ROMS the numbering sequence starts from the inside. The first track is always 'zero'. Tracks are divided into sectors, which store a pre-determined amount of data - usually 512 bytes (4,096 bits).

As the platters are positioned upright on the spindle, the tracks on each side of each platter, which are the same distance from the spindle form an imaginary cylinder. Hardware and software often use the cylinder method for data storage.

Logical formatting

Following physical formatting, the disk is divided into partitions and then logically formatted. This means that a file system is placed on the disk. This file system is used by the operating system, e.g, DOS, Windows95, NTFS, OS/2 to store and retrieve files. If the disk has been divided into multiple partitions, then a different file system could be applied to each partition. Once the logical formatting has taken place, each partition is referred to as a 'volume'.

Partitions

After a disk has been physically formatted, but before the logical formatting, it can be divided into partitions (physical divisions) that are usually spread across both sides of several platters. The parti-

tions (referred to as 'volumes') are given volume names for later identi-
fication.

Many disks are formatted as one large partition. However, it is quite
common to use multiple partitions to:

❖ enable the installation of more than one operating system
❖ make the most efficient possible use of the disk space
❖ make files as secure as possible
❖ physically separate data to make it easy to find any file
❖ make it easy to back up data.

The two main kinds of partitions are known as *primary* and *extended*,
with extended partitions being further subdivided into *logical* partitions.

Primary partitions

A primary partition may contain any operating system, as well as data
files such as user files, applications, etc. The primary partition must be
logically formatted so that it can use a file system that is compatible
with the operating system that has been installed on it.

When multiple primary partitions are created, only one primary parti-
tion may be active (bootable) at a time. This means that the boot se-
quence in another primary partition is not accessible. Most operating
systems can be booted only from a primary partition.

As hard disk drives for desktop PCs move to multi-gigabyte capacities,
the most commonly used PC operating systems cannot manage them as
a single disk partition. To take advantage of this improved storage ca-
pacity, partititioning is often needed to make the most efficient use of
disk capacity.

Extended partitions

An extended partition doesn't directly hold data. It was invented as a
way around the arbitrary four-partition limit on primary partitions.

Logical partitions within the extended partition must be created to hold the data.

Logical partitions

A logical partition is formatted (as its name suggests) logically. Each logical partition can have a different file system thus allowing access to the same files from several operating systems. The only restriction is that the different operating systems must be able to read the file system where the data is stored.

Operating systems that can be booted from a logical partition should be installed in a logical partition leaving the primary partitions free for requirements that actually need a primary partition.

Partitioning and the Boot Process

How a computer boots from a hard disk is governed by how the hard disk is partitioned and the operating system in use.

To start up a computer, the central processing unit (CPU) executes the instructions contained in the ROM BIOS. The procedures, or routine for booting are the last instructions performed. They read into memory the master boot record (MBR) from the first sector of the first physical hard disk. (The MBR consists of a boot program and a table giving details about the disk partitions.)

After the BIOS boot routine is complete, the MBR starts the boot program for the active partition which loads the files for that partition. For hard disks with more than one primary partition, each primary partition has a boot record containing a boot program to start the particular operating system installed on that partition. Primary partition boot records are in the first sector of the partition.

Concepts

Setting Active Primary Partitions

To boot from the hard disk, there must be an active bootable primary partition on the first physical hard disk. When creating multiple primary partitions for different operating systems, the computer must be told which primary partition on the first physical hard disk is to be activated.

Assigning Drive Letters

After booting in DOS the operating system assigns drive letters (e.g, C:, D:) to the primary and logical partition volumes that it recognizes, to allow referencing to the files on the partitions. Drive letters are assigned to primary partitions in the order that they appear on the hard disks, as follows:

C: is assigned to the active primary partition on the first hard disk.

D: is assigned to the first recognized primary partition on the next hard disk, or the next recognizable volume on the first hard disk.

Letters are assigned, in order, to the first recognised primary partitions of further hard disks until all have been assigned drive letters. Drive letters are then assigned to all logical partitions recognised by the operating system, beginning with logical partitions on the first hard disk, and so on, in order. After the logical partitions have been assigned letters, other drives, such as CD-ROM drives) are assigned drive letters.

If there are partitions on hard disks which cannot be recognised by the operating system, they are ignored and the operating system continues assigning drive letters to partitions it recognises.

Drive Letter Changes

An operating system may change those drive letters if a hard disk is added or removed, a partition copies or partitions are reformatted for a different file system. They may also change if booting is done using a different operating system.

When copying partitions it is important to be aware of the possibility that the drive letters may be altered after copying or creating individual partitions.
These changes in a drive letter assignment can result in configuration problems.

Avoiding drive letter changes

When booting using a different operating system, prevent changes to drive letter assignment by placing partitions formatted with file systems that will be recognized by *some* operating systems after files recognized by *all* operating systems. For example, if using Windows NT and DOS, some partitions will be formatted as NTFS partitions and others as FAT partitions. Windows NT recognizes both types, DOS only recognizes the FAT partitions. Therefore if the NTFS partitions are placed after the FAT partitions, the FAT partition drive letters will remain the same no matter which operating system is used for booting. In fact, it is good practice to *always* place FAT partitions first.

File Security and Access

To ensure that files can be read by more than one of the operating systems on the primary partitions, put the files on a logical partition using a file system that can be read by the appropriate operating system.

For easy access to files, organize the files into logical groups within several logical partitions rather than putting all the files into subdirectories under one root directory. Greater security over access to files is also achieved because access can be restricted to certain logical partitions using a file system with good security features.

Using Disk Space Efficiently in FAT Partitions

For a large hard disk which uses a FAT file system for most of the data storage, create multiple logical partitions to contain user files and ensure that the FAT partitions are as small as possible.

The reasoning behind this is that the computer must use a complete cluster to store data no matter how much data is to be stored. The cluster size is determined by the size of the partition, therefore if smaller partitions are used, smaller clusters can be used and there is no wasted space.

For example, a 2Kb file saved to a 1,024Mb FAT16 partition uses a 32Kb cluster – 30Kb of space is wasted.

A partition using a 64Kb cluster wastes a lot of disk space. Therefore it is not advisable to use Windows NT support of 64Kb clusters unless it is absolutely necessary. The partition will also not be stable under any other operating system but Windows NT.

The BIOS 1,024 Cylinder Limit

Some systems have a BIOS 1,024 cylinder limit. This limit applies if the hard disk has more than 540Mb capacity, the BIOS does not have built-in INT 13 BIOS extensions, and/or the hard disk was manufactured before 1994 and does not have LBA (Logical Block Addressing) ability.

If a system has the criteria described above and only DOS is used, the DOS FDISK and *Ghost* will not allow visibility beyond the 1,024th cylinder. This is the case even if an extended partition uses space beyond the 1,024 limit, even if the logical partitions in that extended partition are within the limit.

Operating System Limits

Neither the correct BIOS nor any type of driver can overcome limitations inherent in operating systems. For example, in Windows3.1 and early versions of Windows95, 16 bits are available for cluster addressing in the file allocation table (ergo, FAT16). As each address specifies a cluster of up to 32Kb, the maximum partition size is 2.1Gb. So while hard disk drives larger than 8.4Gb can be used with FAT16 operating systems, multiple partitions no larger than 2.1Gb each are required.

This limitation is overcome in later OEM versions of Windows95, with 32 bits available for addressing in the allocation table (FAT32). FAT32 is included in Windows 98. However, there is no upgrade to allow earlier versions of Windows95 to use FAT32.

The File System

The file system is used to:

❖ track free and allocated space
❖ track where the randomly held parts of a file are physically stored on the disk
❖ maintain the directory and file names.

In other words, the file system contains the organization structure used for efficiently storing and managing data. The structure could include the operating system boot record, the files and the directories.

There are many different file systems, some of which can only be used by one operating system while others are able to be used by a number of different operating systems.

Concepts

File Allocation System

All data stored on hard disk drives relies on some type of file allocation or addressing system, but PC users must understand why the 2.1Gb barrier exists and what choices they have.

For desktop PCs the most common file system used by all Windows and DOS-based machines is FAT. However, FAT has an addressing limitation that prevents it supporting drive partitions larger than 2.1Gb. This limitation does not apply to advanced file systems, such as those used by Windows NT, OS/2 and the Macintosh operating system.

Under DOS 6.x, Windows 3.x and Windows95, disks with capacities of more than 2.1Gb must be partitioned into smaller logical units to be fully accessible by Windows. However, as partitioning can enhance performance, this effort can have additional benefits. The following paragraphs give detailed information about some of the currently available file systems.

FAT

FAT is used primarily by DOS and stores data files in clusters. However, the cluster sizes, which are selected by the DOS format command, are the minimum possible for a given partition size (eg. 8Kb clusters are the smallest possible for a partition size greater than 256 Mb). DOS creates a FAT using 12 bit numbers if the partition is 16Mb or less. FAT clusters can be 512 bytes to 64Kb in size. *Ghost* supports all types of FAT and can copy and resize FAT partitions.

NTFS

NTFS is accessible through the Windows NT operating system and is usually used for greater than 400 Mb disks (basically because its structures take up a lot of space on the disk). Its system structure is based on the master file table (MFT).

With NTFS, data files are stored in clusters (which are not necessarily the same size as the volume, and can be as small as 512 bytes). Using small clusters limits the amount of file fragmentation resulting in less wasted space, accelerated file access, and improved performance.

To guard against data loss, NTFS holds copies of critical parts of the master file table and marks areas of deterioration (bad sectors) so that they are not used. *Ghost* fully understands the NTFS file system and can copy and resize the partitions.

Netware File Systems

The NetWare File System is used by the Novell NetWare networking operating system for network servers. *Ghost* uses a sector-by-sector copy of these partitions.

Linux Native Ext2 can only be used by the Linux operating system. The maximum volume size supported by Linux Ext2 is 4 Terabytes. *Ghost* as at Version 5.0, currently uses a sector-by-sector copy of these partitions.

HPFS

HPFS is also accessible through the Windows NT operating system but is mainly used for OS/2.

Unlike NTFS, HPFS allocates file data in sectors which are monitored for use by organizing volumes into 8Mb bands and 2Kb allocation bitmaps between the bands. This method produces improved performance as the read/write heads do not have to travel back to track zero to pick up information about volume space.

Directories are efficiently organized using filenames, making for quick access and better use of disk space. As at Version 5.0 *Ghost* currently duplicates HPFS on a sector-by-sector basis.

Networking Concepts and *Ghost*

The replication of computer disks and partitions across a network environment link is a logical and useful extension to *Ghost*. *Ghost* allows access to the use of locally mapped network drives, the use of NetBIOS for one-to-one computer connections, and TCP/IP multicasting for fast and efficient concurrent replication of systems. It is important to understand the differences between the options, so the most appropriate and efficient method of replication can be used.

Ghost and Mapped Network Drives

Ghost can take full advantage of networking client software providing mapped network drives. A mapped network drive is a storage resource which the local operating system can access using its drive letter assignments.

For example, a Windows95 machine might have one hard disk installed, with three partitions, assigned to C: D: and E: drive. It also could contain a CD-ROM (F:) and any number of other peripheral drives. As well as these local drives, the Windows95 machine has access to shared or accessible drives on other networked machines assigned to drives G: H: and I:. The G: drive connects to the file server's shared directory, the H: drive to the user's personal directory on the file server, and the I: drive is assigned to the *Ghost* images drive on the file server.

When using *Ghost*, *Ghost* can access and write to the mapped network drives, just as if they were local drives. *Ghost*'s file requester will display these drives in the File Locator window under the drives available section.
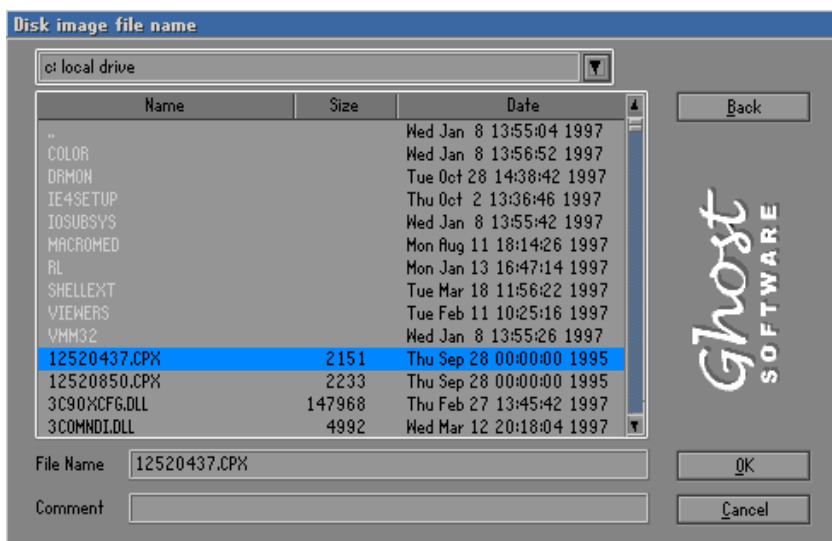
To map a network drive for *Ghost* to use, networking software (usually called network client software) for the DOS operating system needs to be installed and configured. Details on how to set up a mapped network connection are detailed in the Setting Up chapter.

Ghost and NetBIOS

Ghost allows two machines to be connected in a master/slave configuration using a network protocol called NetBIOS. NetBIOS stands for Network Basic Input/Output System. NetBIOS allows one computer running *Ghost* in Master mode to communicate and transfer information with another machine using *Ghost* in Slave mode across a network. NetBIOS can only be used in a one-to-one peer fashion, and does not allow access to file servers and the like.

Concepts

Details on how to set up two machines to communicate together on a peer-to-peer basis is detailed in the Setting Up chapter.

Background Notes on NetBIOS

NetBIOS is a standard originally developed for the IBM PC Network program. Its purpose is to provide a generic set of Application Programming Interface (API) calls that an application developer can use to make two machines communicate together as peers.

The only time that NetBIOS is needed is when you are running an application or driver that specifically calls the APIs that NetBIOS provides. IBM's NetBIOS used to be very popular as a generic means of creating network-aware applications, so there may still be applications available that make use of this API.

The original NetBIOS has several limitations. Primarily, it does not provide a facility for inter-networking, or passing NetBIOS packets across multiple networks. It will only work on a single local network. In other words, IBM's NetBIOS packets cannot cross a Novell router or similar device.

To provide users with the flexibility required, Novell developed a compatible NetBIOS. As far as the application is concerned, it is the same NetBIOS provided by IBM. It provides the same APIs as IBM's NetBIOS, plus a few more.

Novell's implementation of NetBIOS was designed to run on the IPX/SPX protocol. IBM's NetBIOS ran on their own protocol. When Novell designed this implementation of NetBIOS, the ability to traverse inter-network routers was included. Also, where the original NetBIOS' default configuration allows one workstation to have up to six sessions in use simultaneously, Novell's default configurationwill allow 32 sessions (that limit can be increased by using the NetBIOS 'RESET' command).

Because Novell's implementation of NetBIOS is designed to run on IPX/SPX, and other vendors' versions run on other protocols, all nodes wishing to communicate together through NetBIOS must run the same ven-

dor's NetBIOS. That is to say, if one machine is running IBM's NetBIOS and another machine is running Novell's NetBIOS, the application running on one machine will not communicate or see the other machine.

For most implementations, the Novell NetBIOS default configuration is sufficient. However, may configuration options are available through the **net.cfg** file. In that file, you can configure the short and long machine type, timeouts, delays, names, buffers, and so forth. For details on what parameters can be configured in the **net.cfg** see your NetWare documentation.

Ghost and Multicasting

Ghost Multicasting uses TCP/IP multicast packets to send an image file to one or more receiving *Ghost* clients at the same time. *Ghost* Multicasting works over both Ethernet and Token Ring networks.

The idea behind Multicasting is similar to a school classroom. The teacher (Multicast Server) stands up at the front of the room and speaks instructions to the students (Client computers). All the students listen to the teacher at the same time and do what they are told to do. This obviously saves the teacher time by avoiding the need to repeat the instructions to each individual student.

For comparison purposes, in a networking situation using mapped network drives it is also possible to have multiple one-to-one connections running over the same network bandwidth.

In our example, this is like the classroom (Network) having one teacher per student and each student only being prepared to receive instructions from their personal trainer. This now means that each teacher has to share the same room and speak one at a time to the individual students to ensure they can hear them, thus causing the time for a message to get to every student to increase in proportion to the number of students.

With Multicasting, this problem is avoided because only one speech (message) needs to be sent out.

Of course, in some situations, the students misunderstand or miss something the teacher said, and cannot proceed to do what they were told before clarifying what was said. In these cases, the students indicate to the teacher that they missed the sentence, and the teacher will send the details missed. *Ghost* Multicasting includes similar communication techniques, but also has more complicated methods of ensuring that the best attempt to provide the data to all the clients is maintained.

Boot Disks and *Ghost*

Each of the three networking techniques described above needs to be set up differently. The Setting Up chapter introduces the method required for each type of network connection. The methods of setup are not interchangeable and the method of setup taken should reflect the most appropriate option for what you are trying to achieve.

Glossary

ASPI

Advance SCSI Programming Interface. *Ghost* uses an ASPI driver to communicate with SCSI tape devices. ASPI drivers are loaded in the **config.sys** file during bootup in DOS, e.g, **aspi4dos.dos**.

Batch Mode

The use of *Ghost* command line switches to automate the use of *Ghost*, and remove the need for user interaction. Especially useful for large rollouts or repetitive tasks.

BIOS

Basic Input Output System. In some operating systems, the part of the system that customizes it to a specific computer. An interface used by DOS to access hardware.

Boot Disk

A disk containing system files required to boot up an operating system.

BOOTP

Bootstrap Protocol. A TCP/IP network protocol used to configure workstation IP settings. DHCP is an extension of BOOTP.

Boot Sector

The area on a disk containing information required to boot the remainder of the operating system into memory.

Glossary

Broadcast

A network message simultaneously sent to all members of the local network. In TCP/IP terms, broadcast datagrams are sent to all members on the local subnet.

CD-ROM

Compact Disk Read Only Memory. A write-once read-many times removable media useful for storing a *Ghost* image file.

Clone

A duplicate system having the same configuration as the original. The act of replicating.

Cylinder

A parameter that (along with heads and sectors) defines the size of a disk.

Data

A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means.

Destination

The location of the disk, partition or image file that *Ghost* is to write to.

DHCP

An extension of BOOTP. See BOOTP entry.

Disk

A hardware device to store data, typically made up of a Master Boot Record and a number of partitions.

DIX Ethernet

The original Ethernet specified by Digital Equipment Corporation (DEC), Intel Corporation, and Xerox Corporation.

Drive

Another name for a partition. Each drive is accessed by a drive letter, e.g, C:

Ethernet

A media access method using Carrier Sense Multiple Access/Collision Detection (CSMA/CD) used widely in local area networks. Refer also to DIX Ethernet.

Extended Partition

A partition designed to get around the four-partition-per-drive limitation. An extended partition contains one or more logical partitions that data may be stored in.

FAT

File Allocation Table. Used by DOS, Win95 and sometimes OS/2 and Windows NT to allocate space to files and directories.

Ghost

General Hardware Oriented Software Transfer.

Gigabyte (Gb)

Either 1,000,000,000 bytes or, more correctly 1,073,741,824 bytes. *Ghost* uses 1,073,741,824 bytes.

Head

Each disk has a fixed number of heads. A head will read from a single surface of a single platter of a disk.

HPFS

High Performance File System - an alternative to a FAT file system. HPFS is used by OS/2.

Image

See Image File.

Image file

An image file is so named because it contains an exact copy or image of a drive or partition. The image file can then be used for recovery or cloning purposes, to replicate exactly the drive/partition contents.

Interactive Mode

The use of *Ghost* interactive menus to perform required operations.

IPX/SPX

A network transport protocol used in Novell networks, and some MS networks.

Jumper

A changeable hardware switch located on most IDE hard drives to enable drive selection of master/slave and other configuration settings.

Laplink Cable

A special cable that plugs into the parallel ports on two peer computers so that data transfer can occur between them.

Linux

An implementation of the Unix Operating System for Intel machines.

Mapped Drive

A shared drive on another networked computer that has been locally assigned a drive letter.

Master (clone)

When *Ghost*ing using peer-to-peer connections, one computer must be defined as the Master. When cloning from disks or partitions the Master contains the source data. All operator input will be on the Master computer.

Megabyte (Mb)

Either 1,000,000 bytes, or more correctly 1,048,576 bytes. *Ghost* uses 1,048,576 bytes.

Model System

A model system, is the drive/partition being copied that contains the data required for cloning purposes.

Multicasting

A network message simultaneously sent to a group of members of the local network. In TCP/IP terms, multicast datagrams are sent to a group of members listening on a multicast address. Allows concurrent workstation loading while keeping network traffic to a minimum.

Multicast Client

The multicast client machine is the destination machine for the disk/image dump.

Multicast Log File

A file able to be generated by *Ghost* Multicasting for diagnostic purposes.

Multicast Server

The multicast server is the machine containing the disk/image details for dumping on the client machine(s)   .

NetBIOS

A high level Network programming interface that is supported by lower level Network protocols such as IPX/SPX and TCP/IP.

NIC

Network Interface Card.

Glossary

NTFS
New Technology File System. An alternative to FAT and HPFS file systems. NTFS is used by Windows NT.

Partition
A contiguous area on a disk, defined in the Master Boot Record, which has a specific file system such as FAT, HPFS or NTFS.

Partitioning
Subdividing a computer storage area into smaller units that can be allocated to specific jobs or tasks.

Primary Partition
A primary partition may contain any operating system, as well as data files such as user files, applications, etc. The primary partition must be logically formatted so that it can use a file system that is compatible with the operating system that has been installed on it. Most operating systems can be booted only from a primary partition.

Restore
Ghost is used to return a disk/partition to its original state, e.g, for backup purposes.

SCSI
Serial Computer System Interface. (Initially known as Small Computer System Interface). A type of expansion bus generally used to connect hard drives or other storage devices. It is generally used in machines with higher drive performance requirements.

Sector
A 512-byte area on a disk. The smallest addressable unit.

SID
The Windows NT Security Identifier. The SID must be unique for each machine on the network. If you clone a Windows NT machine,

you must make sure each machine has a unique SID to avoid networking problems. The SID can be dynamically assigned by the *Ghost* companion, *Ghost Walker*.

Slave

When using *Ghost* for peer-to-peer connections, one computer must be defined as the Master and the other the Slave. When cloning from image files the Slave contains the source data. See Master.

Source

The disk, disk image file, partition or partition image file used by *Ghost* as the model for cloning to a destination disk or partition.

Span

A term used to indicate the placement of data in an image file across multiple volumes.

Switch

A command line option that allows *Ghost* to be automated and remove the need for user interaction. Commonly used to assist with batch mode operations which are especially useful for large rollouts or repetitive tasks.

TCP/IP

Transmission Control Protocol/Internet Protocol. A networking protocol with broad support deriving from its ability to connect disparate hosts.

Token Ring

A form of ring network that uses token passing to control traffic within the network. Token ring networks are supported by *Ghost.*

Glossary

Track

> The recording surface of each disk is subdivided into concentric areas, or tracks. In one complete revolution of a disk, the read/write head assigned to a disk surface can completely read or write the entire track over which it is positioned.

UTP

> Unshielded Twisted Pair. A type of network cable.

Volume

> A physical unit of a storage medium such as a diskette, hard disk, or disk pack capable of having data recorded on it and subsequently read.

Appendix **A**

Ghost **Command Line Switches Reference**

Ghost may be run in interactive or in batch mode. Batch mode is useful for automating installations for backups using *Ghost*. Most of the *Ghost* switches are used to assist with batch mode operation. To list switches from *Ghost*, type `ghost.exe -h`.

Note that all switches apart from @argument.fil must be preceded with a hyphen (-) or a slash (/). Switches are not case sensitive - they may be entered in upper, lower or mixed case. A detailed list of switches with explanations follows.

@argument.fil

Specifies the file location containing additional command line arguments to be used by *Ghost*. Argument.fil can contain any command line argument, one per line, except for -afile and -dfile switches. The feature allows the command line limit of 150 characters to be exceeded.

<u>Example</u>:

```
ghost.exe @args1.txt
```

-#E=Filename

Standalone switch to bind a *Ghost* license environment file to *Ghost*. Useful when installing or upgrading *Ghost* to a newer version. Refer to Chapter 2 "Getting Started" for further details.

-afile

Overrides the default abort log file with specified file.

-autoname

Autoname spanned dump files.

-batch

Use batch mode. Prevents Abort messages waiting for user acknowl-
edgement, and removes user interaction prompts. Return value must
be checked to identify if ghosting was successful. *Ghost* returns 0 on
success and 1 or higher on failure or error.

-bfc=x

Handles bad FAT clusters. If this switch is set, and the target partition
is FAT, *Ghost* will try to work around bad sectors. The 'x' value indicates
the maximum number of bad sectors allowed to be handled by *Ghost*.
The default value is 500. *Ghost* will abort when a bad sector is
encountered in a non-FAT partition, after the maximum number of bad
clusters is exceeded, or if the switch is not selected.

-blksize=x

Sets the block size to 'x' kilobytes for writing to image files. The block
size can be set from 1kb to 32kb. It is useful in its own right to deter-
mine whether using large buffers across networks is causing problems,
but more importantly it can be used with the VFY switch to allow the
writes to be checked.

-clone

The full syntax for this switch is:

```
-clone,MODE={copy|load|dump|pcopy|pload|pdump
},SRC={drive|file|drive:partition|@MCsessionname|@MTx},
DST={drive|file|drive:partition|@MCsessionname|@MTx},
SZE{F|L|n={nnnnM|nnP|F|V} }
```

Clone using arguments. This is the most useful of the batch switches

and has a series of arguments that define:

a) **MODE**   - This defines the type of clone command to be used:

    COPY   - disk to disk copy
    LOAD   - file to disk load
    DUMP   - disk to file dump
    PCOPY   - partition to partition copy
    PLOAD   - file to partition load
    PDUMP   - partition to file dump

b) **SRC**   - This defines the source location for the operation:

Mode	*Meaning:*
COPY/	
DUMP	Source drive (e.g, **1** for drive one)
LOAD	Disk image filename or device (e.g, g:\Images\system2.img) If tape drive set to **@MTx (x=0...)**
PCOPY/	
PDUMP	Source partition e.g, **1:2** indicates the second partition on drive one.
PLOAD	Partition image filename or device and partition number. Example: g:\images\disk1.img:2 indicates the second partition in the Image file.
Multicasting	**@MC**sessionname where **@MC** indicates multicast and sessionname indicates the session name of Multicast Server sending the required image file.

c) **DST**   - This defines the destination location for the operation:

Mode	*Meaning*
COPY/	
LOAD	Destination drive (e.g, **2** for drive two)

DUMP	Disk image filename or device,
	(e.g, g:\images\system2.img)
	If tape drive set to **@MTx (x=0...)**
PCOPY/	
PLOAD	Destination partition,
	(e.g, 2:2 indicates the second partition on drive two).
PDUMP	Parition image filename (e.g, g:\images\part1.img).
Multicasting	
	@MCsessionname where **@MC** indicates multicast and sessionname indicates the session name for Multicast Server creating image file.

c)   **SZEy**   - Used to set the size of the destination partitions for either a disk load or disk copy operation.

Available y Options:

F   - Resizes the first partition to maximum size allowed based on file system type.

L   - Resizes the last partition to maximum size allowed based on file system type.

n=xxxxM   - indicates that the n'th destination partition is to have a size of xxxx Mb. (e.g, SZE2=800M indicates partition two is to have 800 Mb.)

n=mmP   - indicates that the n'th destination partition is to have a size of mm percent of the target disk.

n=F   - indicates that the n'th destination partition is to remain fixed in size.

n=V   - Indicates that the partition will be resized according to the following rules:

Rule 1 - If the destination disk is larger than the original source disk, then the partition(s) will be expanded to have the maximum amount of space subject to the free space available and the partition type (e.g,

FAT16 partitions will have a maximum size of 2048Mb.)

Rule 2 - If the destination disk is smaller than the original source disk, (but still large enough to accommodate the data from the source disk), the free space left over after the data space has been satisfied will be distributed between the destination partitions in proportion to the data usage in the source partitions. Some examples follow that will help illustrate:

Example 1:
To copy drive one to drive two on a PC, without final prompt if OK to proceed.
```
ghost.exe -clone,mode=copy,src=1,dst=2 -sure
```

Example 2:
To connect via NetBIOS to another PC running Ghost in slave mode, and dump a disk image of local drive two to the remote file c:\drive2.gho
```
ghost.exe -clone,mode=dump,src=2,dst=C:\drive2.gho
-nbm
```
Note: The slave Ghost can be started with ghost -nbs

Example 3:
To copy drive one, second partition on a PC to drive two, first partition on the same PC, without final prompt
```
ghost.exe -clone,mode=pcopy,src=1:2,dst=2:1 -sure
```

Example 4:
To load the disk image file savedsk.gho held on the server drive mapped locally to drive E: onto drive one of the local PC. Do not prompt if OK to proceed.
```
ghost.exe -clone,mode=load,src=E:\savedsk.gho,dst=1
-sure
```

This example is typical of those that would be in a batch file to automate installation of workstations from a network file server.

Example 5:
To dump the second partition of drive one to an image file on a mapped drive g:
```
ghost.exe -clone,mode=pdump,src=1:2,dst=g:\part2.gho
```

Example 6:
To load partition 2 from a two-partition image file on a mapped drive g: onto the second partition of the local disk
```
ghost -clone,mode=pload,src=g:\part2.gho:2,dst=1:2
```

Example 7:
To load drive 2 from an image file and resize the destination partitions into a 60:40 allocation
```
ghost.exe -clone,mode=load,src=g:\2prtdisk.gho,dst=2
,sze1=60P,sze2=40P
```

Example 8:
To clone a three partition disk and keep the first partition on the destination drive the same size as on the source disk, but divide up the remaining space between the other partitions leaving no unallocated space
```
ghost.exe -clone,mode=copy,src=1,dst=2,sze1=F,sze2=V
,sze3=V
```

Example 9:
To load drive one from an image file and resize the first partition to 450 Mb, the second to 1599 Mb and the third to 2047 Mb.
```
ghost.exe -clone,mode=load,src=g:\3prtdisk.gho,dst=1
,sze1=450M,sze2=1599M,sze3=2047M
```

Example 10:
To load a disk from an image file and resize the last partition to its capacity. The first partition utilizes the remaining space.
```
ghost.exe -clone,mode=load,src=g:\2prtdisk.gho,dst=1
,szeL
```

Example 11:
To load drive one from an image file being sent from the multicast

server with the session name "SESSIONNAME" without final
prompt if OK to proceed.

```
ghost.exe -clone,src=@mcSESSIONNAME,dst=1 -sure
```

Example 12:
*To create an image file of drive one to an image file being created
by the multicast server with the session name "SESSIONNAME"
without final prompt if OK to proceed.*

```
ghost.exe -clone,src=1,dst=@mcSESSIONNAME -sure
```

Example 13:
*To create an image file of drive two's partitions to an image file
being created by the multicast server with the session name
"SESSIONNAME".*

```
ghost.exe -clone,src=2,dst=@mcSESSIONNAME
```

Example 14:
*To load drive one from an image file being sent by the multicast
server using session name SESSIONNAME and resize the first par-
tition to 450 Mb, the second to 1599 Mb and the third to 2047 Mb. This
is required to be done in a batch file and under no circumstances
should user intervention be required if a problem occurs. The batch
file commands should alter depending on Ghost 's completion.*

Batch File Contents:

```
@ECHO OFF
ghost.exe -clone,src=@mcSESSIONNAME,dst=1,sze1=450M
,sze2=1599M,sze3=2047M -batch
IF ERRORLEVEL 1  GOTO PROBLEM
ECHO Ghost exited with value 0 indicating success.
REM ** Add any commands required to be run if Ghost
REM    succeeds here**
GOTO FINISH

:PROBLEM
ECHO GHOST returned with an Error value 1 or higher.
ECHO Ghosting was not completed successfully
```

```
REM **Add any commands required to be run if Ghost
REM   fails here**

:FINISH
ECHO Batch File Finished.
```

-CRC32

The -CRC32 switch allows making a list of the files on a disk or partition or in a dumpfile with CRC values for each, and to verify that list against the original or a clone. The purpose is to allow both quick listing of the contents of a dumpfile and verification that a *Ghost*ed disk contains the same files as the original. CRC checking works file-by-file with FAT partitions. NTFS partitions are CRC checked within a dump file by each MFT table. It is not possible at present to obtain a list of files failing a CRC check with an NTFS file system. When a CRC file is created for an NTFS partition, only a single CRC value is generated. You can also create a CRC file from a dumpfile, and verify against a disk.

The full syntax for this switch is:

-CRC32,action={**create**|**verify**|**pcreate**|**pverify**|
dcreate|**dverify**}**,src=**{{Disk Spec}|{Part Spec}|
{File}}**,**{**crcfile**={File}|**vlist**={File}|**vexcept**=
{File}}

Note that no spaces are allowed in the command line.

```
crcfile={File}   :: ASCII CRC file         - default=ghost.crc
vlist={File}     :: Verification list file  - default=ghost.ls
vexcept={File}   :: Verification exception file - no default
```

The possible actions (with descriptions) are:

create
Create an ASCII CRC file from a disk

verify
Verify a disk from a CRC file

pcreate
Create an ASCII CRC file from a partition

pverify
Verify a partition from an ASCII CRC file

dcreate
Create an ASCII CRC file from a dump file
dverify
Verify a dump file from an ASCII CRC file

Examples of -CRC32 in action:

```
ghost.exe -fcr
```
creates a CRC file (called **ghost.crc**) while making a dumpfile.

```
ghost.exe -CRC32,action=create,src=1,crcfile=ghost.crc
```
creates a list of files and CRC values for a disk.

```
ghost.exe -crc32,action=dverify,src=x:dumpfile.gho,
crcfile=ghost.crc
```
verifies the list against a dumpfile.

```
ghost.exe -crc32,action=create
```
creates an ASCII CRC32 file from the primary hard drive. Note that the default disk is the primary drive, the default ASCII CRC32 file is **ghost.crc**.

```
ghost.exe -CRC32,action=create,src=2,crcfile=myfile.txt
```
creates an ASCII CRC32 file. Same as previous except you specify the disk and ASCII CRC32 file. This example uses disk 2 as the source drive and the outfile **myfile.txt.**

```
ghost.exe -CRC32,action=verify
```
verifies the contents of the primary drive against CRC32 file. Once again, the default disk is the primary drive and the default ASCII CRC32 file is **ghost.crc** (in the current directory). In addition the default verification list file is **ghost.ls.**

```
ghost.exe -CRC32,action=verify,src=1,crcfile=myfile.txt,
vlist=myfile.out
```
verifies the contents of the primary drive against CRC32 file. Same as previous but specify the disk, CRC file and list file. This example uses disk 1 as the source drive, **myfile.txt** as the ASCII CRC32 file and **myfile.out** as the verification list file.

```
ghost.exe -CRC32,action=verify,src=1,
crcfile=myfile.txt,vlist=myfile.out,vexcept=myfile.exc
```
verifies the contents of the primary drive against CRC32 file. Same as above with the inclusion of the EXCEPTION argument that excludes compared files based upon its entries.

The VEXCEPT argument specifies files that are not checked with CRC. This is normally used to exclude files that are always changed on boot. A sample exception file follows:

```
[ghost exclusion list]
\PERSONAL\PHONE
[partition:1]
\WINDOWS\COOKIES\*.*
\WINDOWS\HISTORY\*
\WINDOWS\RECENT\*
\WINDOWS\USER.DAT
\WINDOWS\TEMPOR~1\CACHE1\*
\WINDOWS\TEMPOR~1\CACHE2\*
\WINDOWS\TEMPOR~1\CACHE3\*
\WINDOWS\TEMPOR~1\CACHE4\*
[partition:2]
*\*.1
[end of list]
```

The exclusion list is case-sensitive; all files should be specified in upper case. The *wildcard follows Unix rule, it is more powerful than the MS-DOS *. In particular it matches the . as well as any other character, but other characters can follow the *. Thus a wildcard of *br* will match any files containing the letters "br", e.g, brxyz.txt, abr.txt, abc.dbr.

The specification of \WINDOWS\COOKIES*.* in the example above means "match all files in the subdirectory \WINDOWS\COOKIES that have an extension. To match all files with or without an extension, WINDOWS\COOKIES* should be used.

Short filenames should be used in exclusion files.

Files specified before the first [Partition:x] heading will be used to match files in any directory.

A directory of * matches any subdirectory, regardless of nesting. The above exclusion file will match any file with an extension of .1 in any subdirectory on the second partition. Apart from this, wildcards should be used for files, not for directories.

-crcignore
allows restoring of dump files that contain some corrupted files.

-dd
dumps disk metrics to the dump log file **ghststat.dmp**.

-dfile
overrides default dump log file with specified path. (See -dd above.) Not a valid option in @argument.fil.

-di
displays Diagnostics. This is useful for debugging purposes. For each *disk* present on the PC, the physical attributes such as ***Drive***, ***Cylinders***, ***Heads***, ***Sectors per track***, and ***Total sectors*** are displayed. For each *partition* present on each *disk*, the ***Number***, ***Type***, ***Physical/Logical flag***, ***Starting sector*** and ***Number of sectors***

are displayed. The diagnostics may be redirected to a file and emailed or faxed to Technical Support to assist with problem solving.

Example:
```
ghost.exe -di>diag.ls
```
will output disk diagnostics to the file **diag.ls**.

-dl=n

specifies the highest BIOS fixed disk slot to attempt to detect. solves problems where some Pheonix BIOS based systems may hang when Ghost attempts to detect disks or when all disks do not appear in the Ghost display. Valid values for n are 128 to 255.

-drb

disables image file read buffering.

-dwb

disables image file write buffering.

-f12

allows resizing of FAT12 partitions when -**clone** switch is used.

-f64

allows *Ghost* to resize FAT partitions to be greater than 2.1Gb. This is only supported onWindows NT. *Do not use on systems using other operating systems.*

-fatlimit

prevents *Ghost* from resizing FAT partitions beyond 2 Gb. Useful when Windows NT OS Partitions are present on the disk.

-fcr

creates a CRC file (called **ghost.crc**) while making a dumpfile.

-ffd

disables the use of the BIOS IDE FDPT (Fixed Disk Parameter Table) during disk geometry detection and calculation. Some machines' BIOS

contain the ability to select which of the available IDE drives will be used to boot the system. When this BIOS feature is used, it may interfere with *Ghost*'s geometry detection. Use this switch to work around incorrect disk geometry encountered in *Ghost* when the system has been booted fron a second or subsequent IDE drive.

-fff

Forces use of the BIOS IDE FDPT (Fixed Disk Parameter Table) during geometry detection and calculation. When machines have both SCSI and IDE drives installed, the *Ghost* geometry detection may be unable to differentiate between the SCSI and IDE drives installed. This switch can be used when problems are experienced when both IDE and SCSI drives are insalled on the machine, and should only be used when booting from an IDE drive.

-ffg

Overrides the detection of attempts to write beyond a disks capacity. Useful when the geometry detection has detected the drive to be smaller than it actually is.

--finger

displays the fingerprint details written on a *Ghost* ed drive. The fingerprint displays the *Ghost* process used to create the drive or partition and the time, date and disk the operation was performed on.

-fnf

disables the creation of a fingerprint when *Ghost* ing disks or partitions.
Similar to the functionality environment switch FPRNT=N.

-fnw

disables writing to disks or partitions.
Similar to the functionality environment switch WRITE=N.

-fnx

disables Extended INT13 support.

-fro

forces to continue cloning even if source contains bad blocks.

-fx

flag Exit. Normally when *Ghost* has finished copying a new system to a disk, it prompts the user to reboot with a "Press Ctrl-Alt-Del to Reboot" window. However, if *Ghost* is being run as part of a batch file it is sometimes useful to have it just exist back to the DOS prompt after completion so that further batch commands may be processed. -**fx** enables this. See -**rb** for another option on completing a clone.

-h

displays *Ghost* switch Help pages.

-ia

Image All. The Image All switch forces *Ghost* to do a *sector by sector copy* of all partitions. When copying a partition from a disk to an image file or to another disk, *Ghost* examines the source partition and decides whether to copy just the files and directory structure, or to do an image (sector by sector) copy. If it understands the internal format of the partition it defaults to copying the files and directory structure. Generally this is the best option, but occasionally if a disk has been set up with special hidden security files that are in specific positions on the partition, the only way to reproduce them accurately on the target partition is via an image or sector-by-sector copy.

-j=session

replaced with clone @MC switch, refer to -CLONE switch for futher details.

-js=n

set the maximum number of router hops *Ghost* is allowed to cross in an attempt to find the Multicast server. (Default is 10).

-jl:x=filepath

create Multicast diagnostic log file patch at level x.

-lpm

LPT master mode. This switch causes *Ghost* to automatically go into LPT master mode, and is the equivalent of selecting the LPT Master option from the Connection type menu. See Peer-to-Peer connections, Connecting via LPT in Chapter 2 "Setting Up" for more information.

-lps

LPT slave mode. This switch causes *Ghost* to automatically go into LPT slave mode, and is the equivalent of selecting the LPT slave optionfrom the Connection type menu. See Peer-to-peer connections, Connecting via LPT in Chapter 2 "Setting Up " for more information.

-memcheck

Diagnostic memory dump for technical support issues.

-nbm

NetBIOS master mode. This switch causes *Ghost* to automatically go into NetBIOS master mode, and is the equivalent of selecting the NetBIOS Master option from the Connection type menu. See Peer to peer connections, Connecting via NetBIOS in Chapter 2 "Setting Up" for more information.

-nbs

NetBIOS slave mode. This switch causes *Ghost* to automatically go into NetBIOS slave mode, and is the equivalent of selecting the NetBIOS slave option from the Connection type menu. See Peer-to-peer connections, Connecting via NetBIOS in Chapter 2 "Setting Up" for more information.

-nofile

disables the dumpfile dialogue box. For personal preference only. Also useful when opening directories with large numbers of files and

overly slow links.

-ntd

enables NTFS Internal diagnostic checking.

-ntc-

disables contiguous run allocation.

-ntn

inhibits CHKDSK on NTFS volume boot.

-ntic

ignores the NTFS volume CHKDSK bit.

-ntil

ignores non-empty NTFS log filer check (inconsistent volume).

-ntx:y

specifies *Ghost*'s NTFS volume memory cache to be y Kb in size.

-or

OVERRIDE, allows the user to override *Ghost* internal space and integrity checks.

-pwd and -pwd=x

specifies password protection to be used when creating an image file. x indicates the password for the image file. If no password is given in the switch *Ghost* will prompt for one.

-quiet

quiet mode. Disables status updates and user intervention.

-rb

reboots after finishing a load or copy. After completing a load or copy operation, the target PC must be rebooted so that the operating sytem

can load the new disk/partition information. Normally *Ghost* prompts the user to "Press Ctrl-Alt-Del to reboot". **-rb** tells *Ghost* to automatically reboot after completing the clone, and is useful when automating *Ghost* in a batch command file. See the -**fx** switch for another option on completion of a clone.

-skip=x
skips the file 'x' during a disk to image file operation. Wildcards and multiple skips are not supported. For example: -skip =BIGFILE.TMP will skip BIGFILE.TMP when creating the image file.

-sleep
this is useful when using *Ghost* with a mapped file server volume that is experiencing corrupt image files. *Ghost* will often find problems in networks on which other applications run without problems, because *Ghost* transfers a lot of data at once. Using -sleep=10 will slow *Ghost* down and improve the network reliability in this situation. This switch is not intended to be used with Multicasting.

-span
enables spanning across volumes.

-split=x
splits image file into 'x' Mb spans. Use this to create a 'forced' size volume set. For example, if you would like to force smaller image files from a 1024 Megabyte drive, you could specify 200 megabyte segments. For example,
```
ghost.exe -split=200
```
will divide the image into 200 Megabyte segments.

-sure
use the -sure switch in conjunction with -clone to avoid being prompted with the final "Proceed with disk clone- destination drive will be overwritten?" question. This command is useful in batch mode.

-tapebuffered

default tape mode. Sets the ASPI driver to report a read/write as successful as soon as the data has been transferred to it. Useful when using older or unreliable tape devices or sequential media.

-tapesafe
sets the ASPI driver to report a read/write as successful only when the data has been transferred to the physical medium. Useful when using older or unreliable tape devices or sequential media.

-tapespeed=x
allows control of tape speed. Where x equals 0 to F. 0 is default, 1-F increase speeds. Only use this when the tape does not work correctly at the speed used by *Ghost*.

-tapeunbuffered
sets the ASPI driver to report a read/write as successful only when the data has been transferred to the tape drive. (It is possible that this occurs before the data is actually physically written to the medium.)

-ver
displays version number of *Ghost* running.

-vdw
If this switch is set, *Ghost* will use the disk's verify command to check every sector on the disk before it is written. The action *Ghost* takes if a sector fails the verify, depends on the -bfc switch.

-wd-
disables disk caching on destination disk.

-ws-
disables disk caching on source disk.

-xint13on
this forces the use of BIOS extended int13 system calls if present.

-z

Compress when dumping a disk or partition to an image file.

-z or -z1	low compression	(Fast)
-z2	high compression	(Medium)
-z3 thru -z9	higher compression	(Slower)

Appendix A

Appendix **B**

Ghost **Multicast Server Command Line Switches**

The switch references covered in this Appendix are:

❖ *Ghost* Windows Multicast Server command line options
❖ *Ghost* DOS Multicast Server command line options.

Ghost **Windows Multicast Server Command Line Options**
The command line interface is as follows:

ghostsrv filename session [**-n**client_count][**-t**time]
[**-l**{**A**|**I**|**W**|**S**|**E**}][**-f**logfile][**-c**][**-d**][**-p**]

where:

filename =   path and filename of disk image file
session =   session name to use

and optionally:

-n = starts sending to session automatically after client_count c l i -
    ents have joined the session
-t = starts sending to session automatically after specified time (24

hour hh:mm format)

-l = creates log file with log level specified (E,S,W,I, or A)

-f = specifies the logfile name, (by default "**ghostlog.txt**")

-c = closedown **ghostsrv** after multicast session completion

-d = use dump from client mode (load to client is default)

-p = specifies partition mode operation. If loading to clients, the partition number must be given. If dumping from client no partition number is required.

Ghost **DOS Multicast Server command line options**

The command line syntax is:

```
dosghsrv filename session [-l[A|l|W|S|E]]
[-nclient_count][-tstart_time][-d][-p]
```

where:

filename = path and filename of disk image file
session = session name to use

and optionally:

-l = creates log file **rmllog.txt** with the log level specified (E,S,W,L or A)

-n = starts sending to session automatically after the specified client_count clients have joined the session

-t = starts sending to session automatically after the specified 24-hour time occurs

-d = use dump from client mode (load to client is default)

-p = specifies partition mode operation. If loading to clients, the partition number must be given. If dumping from client no partition number is required to be specified.

Appendix **C**

Upgrading Win95's Winsock to Version 2.x

The Windows *Ghost* Multicasting server, **ghostsrv.exe**, uses Winsock 2 to provide the multicasting features in Windows95, Windows98, and Windows NT.

Winsock 2 is currently included in the distribution of Windows NT4.

There are no additional installation steps required to install **ghostsrv.exe** in Windows NT and Windows NT4. However, to run the *Ghost* Multicast server, **ghostsrv.exe** on a Windows95 machine, the following steps are required:

1. You can acquire Winsock2 by pointing your browser to http://www.ghosthelp.com/downloads/multidl/ws2setup.zip

2. Unzip it and read the **ws2setup.txt** file.

2. Run the update names **ws2setup.exe**.

3. Reboot the Windows95 machine.

The update copies several files into the Windows directories, and updates the networking stack to Winsock2 as detailed in **ws2setup.txt**.

Appendix C

<u>Notes</u>:

Caution: *If you are running Windows NT or Windows98, DO NOT install **ws2setup.exe** or any other after market Winsock2 DLLs.* **Winsock2 is built into Windows98 and Windows NT version 4 and is not available for earlier versions of Windows NT.**

Appendix D

The wattcp.cfg Network Configuration File

The **wattcp.cfg** Configuration file contains the TCP/IP networking configuration details for *Ghost* and **dosghsrv** running in DOS. The **wattcp.cfg** file is not required for the Windows-based *Ghost* Multicast server **ghostsrv.exe**.

The **wattcp** file:

- ❖ specifies the IP address of the machine
- ❖ specifies the Subnet mask
- ❖ allows the setting of other optional network parameters
- ❖ should be located in the same directory as **ghost.exe.**

The key words available in the wattcp.cfg configuration file are:

IP
specifies the IP address of the local machine. Each machine must have a unique IP address. *Ghost* supports the use of DHCP and BOOTP servers and defaults to using them when the IP address is left blank or is invalid. DHCP and BOOTP provide automatic assignment of IP addresses to machines. This allows identical boot disks to be used on machines with similar network cards.

Netmask
specifies the network subnet mask.

Bootpto (optional)
overrides the time-out value (in seconds) for BOOTP/DHCP.

Gateway (optional)
specifies the IP address of the Gateway. This option should be used when routers are present on the network and when BOOTP services cross routers.

Receive_Mode (optional)
overrides the automatically configured packet driver mode used by *Ghost*. The modes, in order of preference are 4, 5, and 6.

Some packet drivers misrepresent their abilities in receiving Multicast information from the network and allow the use of packet receive modes that they do not correctly support. Ideally, the packet driver should be set to mode 4 so that it gives only the Multicast packets intended for the *Ghost* Multicast session to *Ghost*. If the packet driver does not support this mode, mode 5 can be used to collect all Multicast packets and give them to *Ghost* to filter. The final option, mode 6, configures the packet driver to give *Ghost* all packets being sent on the network.

Notes:
Comments in the file start with a **;**
Options are set using the format: option = value

Example:
```
receive_mode=5            ;set receive mode
```

Appendix **E**

Frequently Asked Questions (FAQs)

Q. I'm running Norton Anti-Virus. NAV comes up with an error on boot-up.. I select "Repair".   After that the machine does NOT boot and the partition information is wrong. The hard drive is corrupt. What happened?

A. *Unfortunately the selection for this is a bit misleading. The proper selection is "Innoculate". The "Repair" option tries to restore the native partition size and FAT information. The "Innoculate" option deletes the original NAV FAT table and rebuilds it fresh. Choose "Innoculate", not "Repair".*

Q. I go to launch *Ghost* and it reports the error "Bad GET Parameter...". What does this mean?

A. *If you have two drives in the system, be certain that only one is master, and one is slave. If FDISK fails to load, the master/ slave relationship is NOT proper. If you have just one drive in the system, ensure that all cables are firm and that FDISK can load and see the drive properly.*

Q. Do I run *Ghost* inside the operating system, such as Win95, NT or OS/2?

A. *At times Ghost will run inside of the O.S., however it is not best to run Ghost inside Win3.x, Win95, NT or OS/2. This could cause Ghost to fail with Error Messages, or misrepresent the hard drive specifications*

Q. If I shouldn't run *Ghost* inside the OS, how should I launch *Ghost*

A. *It is best to execute Ghost at the "true DOS" level, not a DOS window inside the O.S.. Hitting F8 at "Starting Windows95..." works well, or you can create a bootable diskette, and then launch GHOST*

Q. I know I should launch *Ghost* outside the OS, but then I don't have access to the Network, JAZ, ZIP or CD-ROM drive for saving and loading disk images. How do I work around this?

A. *Create a bootable diskette with nothing on it but the DOS-based drivers or network stack required for access to these devices.*

Q. What is the trial or demonstration version for?

A. *For testing Ghost, determining what it does, and how it works. The trial is NOT for commercial gain or profit. It is not to be used as a migration tool, it is for evaluation only.*

Q. How is *Ghost* priced and licensed?

A. *Ghost licensing is dependent on your intended use for the product. We have developed a range of Licensing options to suit a variety of needs and uses. Many Licenses are based on a per seat concept. That is, every unique workstation which will receive the Ghosted*

image must be covered by your License. For example, if you use Ghost to prepare 200 workstations for Windows95, you must purchase a 200 machine License.

"Per seat" Licenses usually have a 25 machine minimum. The concept of the more you License the less the cost per seat applies.

There are also a range of Unlimited Licenses available for specific requirements and industries.

Call 1-888-446-7898 or 1-800- 817-5119 (USA) or refer to the Web site for contact details of your local Dsitributor.

Q. Can Ghost compress an image file?

A. *Yes. Ghost includes several levels of compression offering a range of performance and storage gains.*

Q. I just downloaded the trial, launched Ghost, got an error message stating "Invalid System Date". What does this error mean?

A. *Ghost trial versions check system date and all files to verify the trial time period. There is most likely a file, or set of files, beyond the trial timeout range. For example, if the trial Ghost times out on March 1st, 1998 and Ghost finds a file beyond that date (May 13th, 1998), it will timeout. The only way around this error is find the file and "touch" it or delete it. The registered version does not check file dates.*

Q. Does Ghost support writing directly to a SCSI tape drive?

A. *Versions 3.2 and greater support writing directly to SCSI tape devices*

Q. I'm using *Ghost* to save an image file up to a server. I'm using a boot disk as instructed. I'm running TCP/IP. *Ghost* takes a long time to save and load an "image" to and from the server. Why?

A. *Ghost rides the network layer or "stack" created. Not all stacks work the same as the next. Ghost only goes as fast as the network layer that you've created. The stack may work well normally, saving files normally. Ghost will really ride the stack aggressively. Try experimenting with different clients.*

Q. Does *Ghost* resize FAT32 partitions?

A. *Yes.*

Q. I'm using a Xircom Pocket Ethernet adapter, and *Ghost* fails. Why?

A. *Ghost supports parallel port cloning, as such, you may lose your network connection through the parallel port network adapter on PE1 or PE2. PE3 seems to be a better network adapter for stability. Drivers and BIOS levels on these devices may also play a role in stability. Try using PCMCIA for faster, more stable connections.*

Q. When will *Ghost* support resizable NTFS partitions?

A. *Versions 3.1 and greater support NTFS resizing*

Q. Does *Ghost* support all protocols?

A. *For Network Clients: Ghost supports ArcNET, Ethernet and To-ken Ring, TCP/IP, IPX/SPX, NetBEUI. As long as you have the clients loaded, and resources mapped, Ghost can save and load to and from these resources. For Multicasting: Ghost supports TCP/IP.*

Q. Will *Ghost* save and load UNIX and NetWare drives?

A. *Ghost should be able to save and load UNIX and NetWare volumes. However, resizability will not be available, and hardware must be the same.*

Q. After cloning and restarting with 3-Com NIC cards with Windows95.. Windows95 keeps finding a new NIC card. The NIC card is the same as on my model machine. Why?

A. *Plug and Play, at times will "see" and "find" devices twice or more. To avoid this, remove the device and all the protocols from the model BEFORE saving the image or cloning. After cloning, re-start, Windows95 will "See" and "Find" the card for the first time, then add the clients, etc.*

Q. Does *Ghost* allow me to specify the saving and loading partitions?

A. *Versions 3.1 and greater support partition selection.*

Q. Does *Ghost* support Macintosh?

A. *No.*

Q. I accidentally specified the wrong cloning, is there any way to restore the original contents?

A. *No. Ghost completely overwrites the target Hard Drive, be careful of your selection of targets*

Q. Does *Ghost* support spanning multiple JAZ or ZIP drives?

A. *Yes.*

Q. *Can I use Windows NT 4.0 Client Administrator to create a multicast boot disk?*

A. No. A TCP/IP boot disk created using the NT 4.0 Client Administrator is **NOT** compatible with *Ghost* Multicasting. *Ghost* includes it own internal TCP/IP stack which cannot be run while another TCP/IP stack is loaded. Steps required to make a Multicast Boot Disk are included in the Setting Up chapter

Q. *Why does Ghost need a Packet Driver for Ghost Multicasting?*

A. *Ghost* 's internal TCP/IP stack uses a *packet driver* to communicate to the network card. The Packet Driver specification allows Software writer in DOS to access the Network card capabilities. The multicast documentation outlines two methods for installing a *packet driver* so *Ghost* multicasting can be used.

Q. *Which packet driver setup option is best to use with Ghost Multicasting?*

A. The two documented options are:
   1.   NDIS   drivers   with   a   *packet   driver*   shim.
   2.   Network card dependent *packet driver*.

   Option 1 - Network card dependent *packet drivers*
   Network card dependent *packet drivers* require less effort to set up. They are not always supplied with some network cards. Some *packet drivers* are not completely compatible with multicasting and require additional configuration to work correctly.

   Option 2- NDIS drivers and a packet Driver Shim
   NDIS drivers are included with network cards more often than *packet drivers*. The setup of a NDIS boot disk currently requires more steps to be carried out.

Q. *What is a Packet Driver Shim and why do I need it?*

A. There are several types of drivers available for network cards. These include the Microsoft/3Com defined NDIS drivers, Novell ODI drivers, and Packet Drivers as well as several others. A Packet Driver shim allows the user to use a non-"packet driver" driver to provide access the network interface cards services and uses this driver to give *Ghost* Multicasting and other applications a packet driver interface to communicate to.

Q. *The Multicasting option is greyed out in Ghost. Why can't I use Ghost Multicasting?*

A. The Multicasting option will not be available if there is no packet driver interface setup on the computer *Ghost* is running on. Alternatively, the option is disabled when the version of *Ghost* being used, or the license of *Ghost* , does not include *Ghost* Multicasting as an available feature.

To enable *Ghost* Multicasting, set up the System to have a packet driver interface installed as described in the Setting Up chapter, or contact your local distributor for further details on upgrading to a Multicast license.

Q. *I have set up the packet driver for my Network Interface Card (NIC) and Ghost is unable to contact or connect to the Ghost Multicast Server. What do I do next?*

A.
1. Ensure you are using the latest version of the *Ghost* Multicast Client and *Ghost* Multicast Server. Check that they are both the same version.

2. Check that all NIC's and cables are correctly connected.

3. Check that the NIC has been set up correctly using the setup program included with the NIC.

4. Check the setup of the driver. Read the documentation for the driver you are using, and note if there are any special options needed to be used for your network setup.

5. Check the **wattcp.cfg** file and/or DHCP/BOOTP Settings. Check all TCP/IP configuration settings. Ensure both the server and client have valid IP addresses and are on the same subnet with the correct subnet mask. If routers are present between the client and server, ensure the gateway is set up on the local subnet and is specified in the **wattcp.cfg** file or the BOOTP/DHCP response.

6. If you are running the *Ghost* Multicast Server on Windows95, and you have not already installed the Winsock 2 update, then do so now.

7. Start the *Ghost* Multicast Server and set up the session name and file name for the multicast session. Press accept clients.

8. Start the *Ghost* Multicast Client and attempt to contact the multicast session. Check the session names are identical.

9. If this fails, retest with an alternative receive mode of the Packet Driver. Add the line RECEIVE_MODE=x in the **wattcp.cfg** file. (where x can be 4, 5 or 6. 4 is default.) Refer to Appendix D for more details.

10. Try an alternative packet driver setup for the NIC. For example, if you are trying to use the NIC's packet driver, then set up the NDIS 2.0.1 DOS driver and the *Ghost* supplied NDIS Packet Driver shim.

11. Start the *Ghost* Multicast server and *Ghost* Multicast client with full multicast logging (ALL logging setting). For details on how to do this, refer to Chapter 5, "Multicasting". These logs will assist in further diagnosing the problem.

12. If you require further assistance, contact *Ghost* Technical Support.

Appendix **F**

Troubleshooting

Procedure:

- ❖ Verify that you are running the latest version. Many problems have been resolved through *Ghost* revisions.
- ❖ Browse through the Frequently Asked Questions (FAQs) and Error Codes to find solutions for the more common problems.
- ❖ Search the interactive *Ghost* Software Forum at http://www.ghosthelp.com for questions that may have already been answered.
- ❖ Submit a problem report sheet at the GhostHelp web site or send an email to support@ghosthelp.com explaining the nature of your problem.

Call *Ghost* Technical Support at (1-414) 964-4099.

Note: If you are sending email to *Ghost* Technical Support, make certain that your email program has your correct return address so that you can receive a reply.

Appendix F

Ghost **Error Codes**

Below is a list of the more common errors that *Ghost* may report. Make sure you are running the latest version as many errors have been fixed through *Ghost* revisions. For the latest list of error codes visit the website http://ww.ghosthelp.com.

10000
Incorrect path/file syntax. Ensure path and filename are correct and complete. Also make sure you have the proper user rights if trying to create an image file on the network.

10030
Your Multicast Client is incorrectly configured. Please check your multicast configuration.

10060
Bad read from resource.
Network - Due to traffic or collisions.
CDROM - Due to media problem.

10082
The Demo version of *Ghost* has timed out. It's time to buy!

10170
Use **ghost -or** to override the check or update to the latest version which resolves this problem.

10180
Hard drive is not responding. Check cabling, power connections, jumpers and BIOS setup. Make sure your system recognizes the hard drive by running FDISK.

10210
Invalid extended partition information probably due to the use of disk overlay software like EZ-Drive. *Ghost* cannot address drives entirely if they are being controlled by this type of driver.

10220
Most commonly associated with executing *Ghost* in a Windows DOS box. Execute *Ghost* from a DOS prompt, preferably from a DOS boot diskette.

10600
Ghost cannot continue properly due to lack of memory. See the 15040 error below for more information.

12080
Commonly associated with attempting to clone disk-to-disk over a network. *Ghost* will only clone disk-to-disk on a network via the NetBIOS (peer to peer) convention.

12090
A physical read or write error from or to the hard drive. Try Scandisking with a full sector scan.

14030
An unregistered version of *Ghost* has encountered a file with a date beyond its expiration date. Scan your system for files beyond this date and temporarily remove them from the system to allow *Ghost* to continue. You can locate the offender by looking at the drive:\path\filename at the bottom of the *Ghost* window when this error occurs.

15010 - 15020 - 15030 - 15050
Usually due to problems dealing with the EA DATA. SF file. Execute **ghost.exe -e** to get around this error.

15040
Ghost cannot execute properly due to lack of memory. Make sure you're loading upper memory drivers in your **config.sys** or upgrade to v4.x. Below is an example of a possible config.sys configuration:

```
device=himem.sys
device=emm386.exe noems i=b000-b7ff
dos=high,umb
devicehigh=(your device driver)
devicehigh=(your device driver) etc.
```

Below are devices you DO NOT need. They will not benefit *Ghost*'s performance.

* ❖ setver.exe
* ❖ smartdrv.exe
* ❖ any other disk cache utility
* ❖ mouse.com, or
* ❖ any other mouse driver.

15100
Use `ghost -or` to override the check or update to the latest version which resolves this problem.

15150 - 15160
Probable corrupt image file. Run the "Check image file integrity…" option from the Local/Server menu. Most commonly evident following an image "Drag and Drop" copy. When copying images, open up a DOS window and use the "copy/v x:\image.gho y:\image.gho" to force verification of the copy.

15165
Ghost came across a marginal file usually located in an Internet Browser's cache directory. You can locate the offender by looking at the drive:\path\filename at the bottom of the *Ghost* window when this error occurs. Update to the latest version which resolves this problem.

15170
Due to an unformatted or invalid partition on the source hard drive. Make certain the source drive is completely allocated as *Ghost* looks for 100% viable media.

15175

Older error associated with the Compaq diagnostic partition. Update to the latest version.

16040

Too many partitions on your drive. Resolved in *Ghost* 3.x.

19080

Most likely *Ghost* came accross a directoryname or filename that is corrupt.

19320

Ghost cannot continue properly due to lack of memory. See the 15040 error above for more information.

19330

Due to the incorrect FAT type in your Master Boot Record. Most commonly occurs when your partition is NTFS, but the partition information is still configured as FAT16. You will need to call Technical Support to solve this problem.

12075

Cylinder translation problem most commonly fixed by executing **ghost.exe -fnx**. This could also be caused by the absence of LBA support on drives larger than 528Mb.

SYSTEM ABORT MESSAGES

Below are some system error messages you may encounter. Note that these errors are not caused by *Ghost*, these are errors caused by malfunctioning hardware or software configurations.

```
CDR101: Not ready reading drive X, Abort, Retry, Fail
```

Appendix F

The image file on the CD is not readable. To verify this, try going into DOS and copying the image file off the CD-ROM. Ideas on this can be scanned for and discussed on the *Ghost* Software Forum at www.ghosthelp.com.

Appendix G

Diagnostics

This appendix contains information that may be helpful for diagnostic purposes.

Elementary testing techniques

There are several tools and utilities available in Microsoft's TCP/IP application suite. Here is an example of two tools for the Windows95 server example described in this documentation: **ping.exe** and **winipcfg.exe**. **Ping.exe** and **ipcfg.exe** is available in Windows NT.

Ping shows Computer reply and can be used to show connectivity of Clients. Clients will only respond when they are in Multicast mode. Ping local host shows basic local TCP/IP functionality.

Example: In a Windows DOS prompt dialogue box:

Example 1 - Pinging a Local Host

```
Pinging Win95PC1 [127.0.0.1] with 32 bytes of data:

Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
```

```
Reply from 127.0.0.1: bytes=32 time<10ms TTL=128
```

Example 2 - Pinging a *Ghost* Multicast Client
```
C:\>  Ping 192.168.100.3
Pinging [192.168.100.3] with 32 bytes of data:

Reply from 192.168.100.3:  bytes=32 time<10ms TTL=128
Reply from 192.168.100.3:  bytes=32 time<20ms TTL=128
Reply from 192.168.100.3:  bytes=32 time<20ms TTL=128
Reply from 192.168.100.3:  bytes=32 time<20ms TTL=128
C:\>winipcfg
```

This allows the address range to be 192.168.100.1 to 192.168.100.254 on other machines. Winipcfg verifies the Windows95 IP Configuration Parameters.
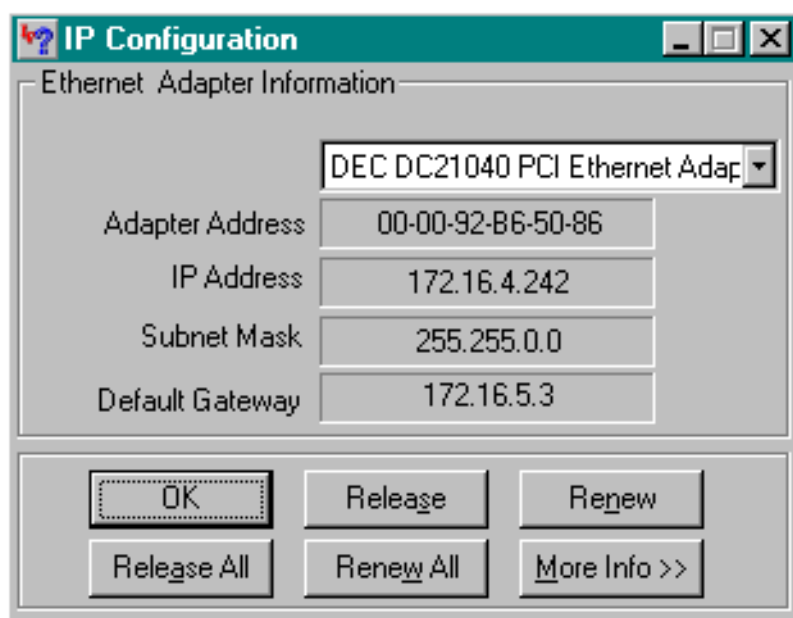
Multicasting Diagnostic and Logging Options

Generating a *Ghost* Multicast Log File

A *Ghost* Multicast log file can be generated for diagnostic purposes. It should be noted that logging slows down the Multicasting process and should be used to assist in diagnosing problems noted during normal use. It is not intended for general use.

Logging in *Ghost* Windows Multicast server, ghostsrv

There are four steps to generate a log file in **ghostsrv**.

1. Select the Options item from the File menu.
2. Select the logging level for the log file.
3. Enter the log file location and name.
4. Use Multicast server application: logging options are engaged.

Step 1 - Select the Options item in the File menu
The Options dialogue is accessible through the File menu.

Step 2 - Select the logging level for the log file
The *Ghost* Multicast log file allows various levels of diagnostic information to be provided. Selection of the logging level is completed through the log level combo box. The diagnostic levels in order of increasing detail are:

❖ Error
❖ Statistical
❖ Warning
❖ Information
❖ All

Step 3 - Enter the log file location and name
Specify a filename and path where the log file should be generated in the Log File box. The Browse button can be used to assist in finding a location for the file. A sample screen is shown below:
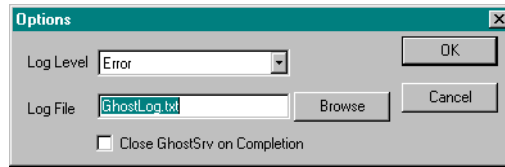
Step 4 - Use Multicast server as required

The *Ghost* Multicast server can be used for normal operation and the log file inspected upon completion.

Logging in *Ghost* DOS Multicast server - dosghsrv

There are two steps to generate a log file while using **dosghsrv**:

1.   Use the Command line switch to activate logging.
2.   Use the Multicast server application.

Step 1 - Use the command line switch to activate logging

To activate *Ghost* Multicast logging, add the logging switch -**lx** where where parameter 'x' specifies the diagnostic reporting level and can be any of the following, E, S, W, I, or A.

Step 2 - Select the logging level for the log file

Use other command line options as required.

<u>Example</u>:

```
dosghsrv.exe -la -n10 c:\test123.gho TestSession
```

starts a Multicasting session called TestSession and uses the file c:\test123.gho. The connecting client's IP address will be displayed on screen. The session transmission is started automatically when 10 clients have connected. A log file **rmllog.txt** will be created for debugging purposes. Please note that using a log file will reduce the performance of the Multicast transmission.

Diagnostic and Logging Options : Client

Generating a *Ghost* Multicast Log File

A *Ghost* Multicasting log file can be generated for diagnostic pur-poses. It should be noted that logging slows down the multicasting process and should only be used to assist in diagnosing problems noted during normal use.

This command-line switch is not intended for general use:

-jl:x = filename. Enable Multicast session logging at selected log level x to file.

where parameter 'x' specifies the diagnostic reporting level and can be any of the following, E, S, W, I, or A.

The diagnostic levels in order of increasing detail are:

❖   Error
❖   Statistics
❖   Warning
❖   Information
❖   All.

To create a Multicasting diagnostic log file from the command line switch

```
ghost.exe -jl:x=d:\filename
```

should be added when starting *Ghost*.

The path should be a location to a drive other than the one being written to by *Ghost*.

Appendix G

Example:

```
ghost.exe -jl:E=d:\logs\multi.log
```

Saves a Multicast log file when multicasting is selected by the user in interactive mode. The log generator reports any errors that occur to the file located at d:\logs\multi.log.

Appendix **H**
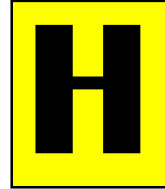
Customizing the Functionality of *Ghost*

Ghost includes the ability for the licensed user to tailor *Ghost*'s functionality provided to the end user. In some situations, the holder of a *Ghost* license may want to provide editions of the *Ghost* executable that has some features disabled, to users within their licensing scheme

To limit the functionality provided in *Ghost*, the *Ghost* license environment file is required. The *Ghost* environment file includes:

❖ the licensed user's details
❖ the maximum number of licensed concurrent users
❖ additional product licensing information
❖ functionality switches.

The *Ghost* executable is configured with the environment file to allow it to be used as detailed in the Getting Started chapter.

The optional switches parameter line in the environment file is the only line that should be altered. Each feature apart from IMGTMO can be activated with `switchname=y` or deactivated `switchname=n` in the bound executable.

The following switches are available:

LOAD
Load disk or partition from image file actions

DUMP
Dump disk or partition to image file actions

WRITE
Stops Ghost from actually writing to destination partition or disk

DISK
Disk-to-disk and partition-to-partition actions

PEER
LPT, NetBIOS and Multicasting options

FPRNT
Creation of fingerprint. A fingerprint is a hidden mark on a cloned drive or partition that details the following:

- the *Ghost* process used to create the drive or partition
- the time the operation was performed
- the date the operation was performed
- the disk number

IMGTMO
Image Time-out Value. Sets the maximum age of an image file in days.

TIMEOUT
Disables *Ghost* until a valid license is reapplied to it.

Examples:

Example 1:
A company may have 100 laptops in use by their sales staff, with the IT System Administrator controlling the organization and maintenance of these laptops. Each of these laptops in use could include a copy of *Ghost*

and model image file burnt in on a CD-ROM for fast system restoration by the users. The System Administrator can configure the *Ghost* edition that is burnt onto the CD-ROM to enable only image file restoration, thus removing the possibility of the end users attempting to use the other functions of *Ghost*.

The Administrator's version of *Ghost* has all options available after binding, using the original environment file, and the CD-ROM *Ghost* version is bound with:

KeyNum	: 12345
License	: BM-512
MaxUsers	: 10
Name	: ABC Inc
Address1	: 200 John Wayne Blvd.
Address2	: Irvine, CA 1024
Switches	: load=y,dump=n,disk=n,peer=n

Example 2:

Ghost can be used as a backup tool. In this case, it may be advisable to disable the load option so that image file creation procedures can be carried out, without the possibility of users accidentally overwriting their local drive. Restoration would require the availability of another executable, or used in conjunction with *Ghost-Explorer*.

Switches   : load=n,dump=y,disk=n,peer=n

Index

A

B

C

Index

Index

Index

Index

Index

User Notes

User Notes

User Notes

User Notes

User Notes