Symantec™

Deploy Toolkit 5.7

User’s Guide

Symantec™ Deploy Toolkit 5.7
User's Guide
Published November 2006
© 2006 Symantec Corporation

WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID, SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Symantec Corporation

20330 Stevens Creek Blvd.

Cupertino, CA 95014 USA

http://www.symantec.com

SYMANTEC SOFTWARE LICENSE AGREEMENT

Symantec Ghost Solution Suite

SYMANTEC CORPORATION AND/OR ITS SUBSIDIARIES ("SYMANTEC") IS WILLING TO LICENSE THE SOFTWARE TO YOU AS AN INDIVIDUAL, THE COMPANY, OR THE LEGAL ENTITY THAT WILL BE UTILIZING THE SOFTWARE (REFERENCED BELOW AS "YOU" OR "YOUR") ONLY ON THE CONDITION THAT YOU ACCEPT ALL OF THE TERMS OF THIS LICENSE AGREEMENT. READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE USING THE SOFTWARE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU AND THE LICENSOR. BY OPENING THIS PACKAGE, BREAKING THE SEAL, CLICKING THE "AGREE" OR "YES" BUTTON OR OTHERWISE INDICATING ASSENT ELECTRONICALLY, OR LOADING THE SOFTWARE, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS, CLICK THE "I DO NOT AGREE" OR "NO" BUTTON OR OTHERWISE INDICATE REFUSAL AND MAKE NO FURTHER USE OF THE SOFTWARE.

1. License:

The software and documentation that accompanies this license (collectively the "Software") is the proprietary property of Symantec or its licensors and is protected by copyright law. While Symantec continues to own the Software, You will have certain rights to use the Software after Your acceptance of this license. This license governs any releases, revisions, or enhancements to the Software that the Licensor may furnish to You. Except as may be modified by an applicable Symantec license certificate, license coupon, or license key (each a "License Module") that accompanies, precedes, or follows this license, and as may be further defined in the user documentation accompanying the Software, Your rights and obligations with respect to the use of this Software are as follows.

If You are using the Software as an end user, as indicated by your license certificate, You may:

A. use the Software for the number of Devices as have been licensed to You by Symantec under a License Module. "Device" is defined as a single computer, storage drive or other device including virtual devices running any of the supported operating systems (i) on which Licensee can install and use the Software, or (ii) from which License accesses and uses the Software installed on the network, or (iii) a physical connection point that links together two separate devices. For devices running multiple operating systems (e.g., multiple boot systems or virtual machine servers) each instance of new operating systems constitutes a new "device" from which the licensee access and uses the Software. Each "device" requires a unique license of the Software. If the Software is part of a suite containing multiple Software titles, the number of copies You may use may not exceed the aggregate number of copies indicated in the License Module, as calculated by any combination of licensed Software titles. Your License Module shall constitute proof of Your right to make such copies. If no License Module accompanies, precedes, or follows this license, You may make one copy of the Software that You are authorized to use on a single computer;

B. make one copy of the Software for archival purposes, or copy the Software onto the hard disk of Your computer and retain the original for archival purposes;

C. use the Software on a network, provided that You have a licensed copy of the Software for each computer that can access the Software over that network;

D. use the Software in accordance with any written agreement between You and Symantec; and

E. after written consent from Symantec, transfer the Software on a permanent basis to another person or entity, provided that You retain no copies of the Software and the transferee agrees in writing to the terms of this license.

If you are using the Software as a value-added reseller, as indicated by your license certificate, You may:

A. use the Software to deploy a single image on the number of Devices as have been licensed to You by Symantec under a License Module. If more than one image is applied to a Device, you are required to purchase a license for every additional image that is applied to that Device. For the purposes of clarity, if "n" images are applied to a single Device, then You are required to purchase "n" licenses.

B. include a disaster recovery image file created by the Software for disaster recovery purposes with any hard drive originally imaged by the Software and transferred to a third party solely for the purpose of allowing You to provide disaster recovery support to such third party. The number of image files transferred to third parties may not exceed the number of licenses held by You, and such use shall not be construed so as to increase the number of licenses granted under this agreement; and

C. provide a copy of a special Ghost executable or PQIDeploy executable that has been modified by Symantec for the purposes of allowing the customer to redeploy the disaster recovery image file onto the hard drive as specified in B above. The special Ghost or PQIDeploy executable has been modified so that it can read images for deployment, but cannot create a new hard drive image; and

D. make one copy of the Software for archival purposes, or copy the Software onto the hard disk of Your computer and retain the original for archival purposes.

You may not:

A. copy the printed documentation that accompanies the Software;

B. sublicense, rent, or lease any portion of the Software; reverse engineer, decompile, disassemble, modify, translate, make any attempt to discover the source code of the Software, or create derivative works from the Software;

C. use the Software as part of a facility management, timesharing, service provider, or service bureau arrangement;

D. use a previous version or copy of the Software after You have received and installed a disk replacement set or an upgraded version. Upon upgrading the Software, all copies of the prior version must be destroyed;

E. use a later version of the Software than is provided herewith unless You have purchased corresponding maintenance and/or upgrade insurance or have otherwise separately acquired the right to use such later version;

F. use, if You received the software distributed on media containing multiple Symantec products, any Symantec software on the media for which You have not received permission in a License Module;

G. ship, bundle, or provide to any third party any version of the Software or transfer any of Your rights to the Software without the prior written consent of Symantec; or

H. use the Software in any manner not authorized by this license.

2. Content Updates:

Certain Software utilize content that is updated from time to time (including but not limited to the following Software: antivirus software utilize updated virus definitions; content filtering software utilize updated URL lists; some firewall software utilize updated firewall rules; and vulnerability assessment products utilize updated vulnerability data; these updates are collectively referred to as "Content Updates"). You shall have the right to obtain Content Updates for any period for which You have purchased maintenance, except for those Content Updates that Symantec elects to make available by separate paid subscription, or for any period for which You have otherwise separately acquired the right to obtain Content Updates. Symantec reserves the right to designate specified Content Updates as requiring purchase of a separate subscription at any time and without notice to You; provided, however, that if You purchase maintenance hereunder that includes particular Content Updates on the date of purchase, You will not have to pay an additional fee to continue receiving such Content Updates through the term of such maintenance even if Symantec designates such Content Updates as requiring separate purchase. This License does not otherwise permit the licensee to obtain and use Content Updates.

3. Limited Warranty:

Symantec warrants that the media on which the Software is distributed will be free from defects for a period of thirty (30) days from the date of delivery of the Software to You. Your sole remedy in the event of a breach of this warranty will be that Symantec will, at its option, replace any defective media returned to Symantec within the warranty period or refund the money You paid for the Software. Symantec does not warrant that the Software will meet Your requirements or that operation of the Software will be uninterrupted or that the Software will be error-free.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE ABOVE WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE AND COUNTRY TO COUNTRY.

4. Disclaimer of Damages:

SOME STATES AND COUNTRIES, INCLUDING MEMBER COUNTRIES OF THE EUROPEAN ECONOMIC AREA, DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE BELOW LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW AND REGARDLESS OF WHETHER ANY REMEDY SET FORTH HEREIN FAILS OF ITS ESSENTIAL PURPOSE, IN NO EVENT WILL SYMANTEC BE LIABLE TO YOU FOR ANY SPECIAL, CONSEQUENTIAL, INDIRECT, OR SIMILAR DAMAGES, INCLUDING ANY LOST PROFITS OR LOST DATA ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF SYMANTEC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

IN NO CASE SHALL SYMANTEC'S LIABILITY EXCEED THE PURCHASE PRICE FOR THE SOFTWARE. The disclaimers and limitations set forth above will apply regardless of whether or not You accept the Software.

5. U.S. Government Restricted Rights:

RESTRICTED RIGHTS LEGEND. All Symantec products and documentation are commercial in nature. The software and software documentation are "Commercial Items," as that term is defined in 48 C.F.R. section 2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are defined in 48 C.F.R. section 252.227-7014(a)(5) and 48 C.F.R. section 252.227-7014(a)(1), and used in 48 C.F.R. section 12.212 and 48 C.F.R. section 227.7202, as applicable. Consistent with 48 C.F.R. section 12.212, 48 C.F.R. section 252.227-7015, 48 C.F.R. section 227.7202 through 227.7202-4, 48 C.F.R. section 52.227-14, and other relevant sections of the Code of Federal Regulations, as applicable, Symantec's computer software and computer software documentation are licensed to United States Government end users with only those rights as granted to all other end users, according to the terms and conditions contained in this license agreement. Manufacturer is Symantec Corporation, 20330 Stevens Creek Blvd., Cupertino, CA 95014, United States of America.

6. Export Regulation:

Certain Symantec products are subject to export controls by the U.S. Department of Commerce (DOC), under the Export Administration Regulations (EAR) (see www.bxa.doc.gov). Violation of U.S. law is strictly prohibited. You agree to comply with the requirements of the EAR and all applicable international, national, state, regional and local laws, and regulations, including any applicable import and use restrictions. Symantec products are currently prohibited for export or re-export to Cuba, North Korea, Iran, Iraq, Libya, Syria and Sudan or to any country subject to applicable trade sanctions. You agree not to export, or re-export, directly or indirectly, any product to any country outlined in the EAR, nor to any person or entity on the DOC Denied Persons, Entities and Unverified Lists, the U.S. Department of State's Debarred List, or on the U.S. Department of Treasury's lists of Specially Designated Nationals, Specially Designated Narcotics Traffickers, or Specially Designated Terrorists. Furthermore, You agree not to export, or re-export, Symantec products to any military entity not approved under the EAR, or to any other entity for any military purpose, nor will You sell any Symantec product for use in connection with chemical, biological, or nuclear weapons or missiles capable of delivering such weapons.

7. General:

If You are located in North America or Latin America, this Agreement will be governed by the laws of the State of California, United States of America. Otherwise, this Agreement will be governed by the laws of England and Wales. This Agreement and any related License Module is the entire agreement between You and Symantec relating to the Software and: (i) supersedes all prior or contemporaneous oral or written communications, proposals, and representations with respect to its subject matter; and (ii) prevails over any conflicting or additional terms of any quote, order, acknowledgment, or similar communications between the parties. This Agreement shall terminate upon Your breach of any term contained herein and You shall cease use of and destroy all copies of the Software. The disclaimers of warranties and damages and limitations on liability shall survive termination. Software and documentation is delivered Ex Works California, U.S.A. or Dublin, Ireland respectively (ICC INCOTERMS 2000). This Agreement may only be modified by a License Module that accompanies this license or by a written document that has been signed by both You and Symantec. Should You have any questions concerning this Agreement, or if You desire to contact Symantec for any reason, please write to: (i) Symantec Customer Service, 555 International Way, Springfield, OR 97477, U.S.A., (ii) Symantec Customer Service Center, PO BOX 5689, Dublin 15, Ireland , or (iii) Symantec Customer Service, 1 Julius Ave, North Ryde, NSW 2113, Australia.

8. Additional Uses and Restrictions:

Notwithstanding any of the terms and conditions contained in this Symantec Software License Agreement, the following additional terms apply to the Software you have purchased:

A. If You are an end user, as indicated on your license certificate, the following uses and restrictions apply:

1. The Software may contain multiple components. For each copy of the Software, such components, whether used individually or collectively, may be used on only one Device.

2. You may:

(i) use the Software only in accordance with its documentation;

(ii) use the Software on one Device to create or deploy an image of a hard drive on that Device, or to another hard drive on the same Device. If a License Module accompanies, precedes, or follows this license, You may make and use the number of copies of the Software set forth in Your License Module on an equal number of individual Devices pursuant to the terms of this License Agreement. Your License Module shall constitute proof of Your right to make and use such copies;

(iii) use the Software on a single Device to create an image file of a hard drive on that Device and store the image file on removable media for disaster recovery purposes;

(iv) use the Software to create a boot disk as described in the documentation for reapplying the hard drive image that was created for disaster recovery purposes to the hard drive on the original Device; or

(v) use the Software to clone a hard drive from a Device to a replacement Device in the manner described in the software documentation, and to use the Software on the replacement Device provided that the Software has been removed from the original Device and the original Device has been Permanently Decommissioned.

(vi) Redeploy the license for the Software from one Device to a new Device, where the original Device is Permanently Decommissioned. Permanently Decommissioned is defined as physically deleting all software, including all operating systems, from the hard drive(s) of the Device.

3. Scripting Documentation:

Any scripting documentation provided in conjunction with the Software may be used only in conjunction with the Software. You may not copy, distribute, license or sublicense the scripting documentation.

4. If your Software includes the Symantec User Migration, the following uses and restrictions apply to the Software:

(i) You may use a single copy of the Software to copy and deploy data and settings on a single Device. Such copy may be used an unlimited number of times for such single Device;

(ii) You may use a single copy of the Software to copy and transfer data and settings from one Device to a replacement Device, provided that the original Device is permanently decommissioned or destroyed; and

(iii) In order to deploy data and settings to multiple Devices, You must have a copy of the Software for each Device that is the recipient of such data and settings.

B. If You are a value-added reseller, as indicated on your license certificate, and Your Software includes the Symantec User Migration, You may use a single copy of the Software to copy and deploy data and settings on a single Device. Once the data and settings have been deployed to that single Device, one license has been consumed. If You deploy a different set of data and settings to that Device, an additional license is required.

Table of Contents

Introduction

Chapter 1: PartitionInfo

Chapter 2: PQAccess

Chapter 3: PQQuery

Chapter 4: PQIExtract

Chapter 5: PQA

Chapter 6: PTEdit

Chapter 7: Virtual Boot Environment (VBE)

Chapter 8: VFILE System Utility

Chapter 9: PQDisk

Chapter 10: PQIDeploy

Appendix A: PQDisk Error Codes

Index

Introduction

The Symantec™ Deploy Toolkit is specifically designed to meet the system configuration and deployment needs of equipment manufacturers, configuration centers, and enterprise IT professionals.

The Symantec Deploy Toolkit can save money by reducing the time required to set up and configure new workstations, deploy software applications across an entire network and upgrade systems to new operating systems and larger hard drives.

What is Included in the Toolkit?

- Partition Information (PartInfo)
- PQAccess
- PQIExtract
- PQA
- PQQuery
- PTEdit
- Virtual Boot Environment
- VFILE
- PQDisk

What Can the Toolkit Do for Me?

The utilities in the toolkit perform a variety of tasks that are common and useful for a configuration center or an IT department. The programs are available for multiple operating systems, so they are easy to integrate with your existing procedures. In addition, many of the programs are scriptable, so if you perform some functions repeatedly, you can create script files to automate the process.

- PartitionInfo is a diagnostic tool the reports the contents of the partition table and boot sector information for each partition. Troubleshooting enables you to find and repair problems in your deployment process, improving reliability.

- PQAccess lets you access files and directories on inactive (hidden) or unsupported partitions, thus eliminating the time necessary to boot to the inactive partition as part of your deployment process.

- PQQuery is a DOS command line program that returns disk and partition information in environment variables. PQQuery allows batch files or imaging scripts to test environment variables and perform different actions depending on the returned values.

- PQIExtract lets you restore individual files or directories from within PQI image files. PQIExtract is useful when you want to restore individual files or reinstall the operating system, but you do not want to lose customized settings or applications that were installed after the image file was created.

- PQA is an image addendum system lets you modify files or directories on a visible or hidden partition on a hard disk. You can use PQA to replace device driver files, provide patch files, rename files, and add or delete files and directories. You build a package of changes, then apply it to the systems you want to change. With PQA, you avoid the necessity of making new image files when you want to change a few files.
- The Virtual Boot Environment eliminates the need for a physical boot floppy or bootable CD to run non-Windows operating systems or applications on Windows systems. You can also use VBE to launch hardware upgrade programs like BIOS flash or other low-level programs from within Windows.
- The VFILE system utility lets you create and modify a bootable floppy image file directly on the computer without needing to first create a physical floppy disk.
- PQDisk is a scriptable version of Norton™ PartitionMagic®. It is intended for use by PC configuration centers and corporations that need to configure large numbers of PCs. It uses ASCII text script files to specify operations that create and manipulate hard disk partitions.
- PQIDeploy is a special version of ImageCenter designed to automate processes in a configuration center or system builder production line. It contains the full ImageCenter scripting capability. PQIDeploy can help you save time by eliminating the need for user intervention.

System Requirements

- IBM PC or compatible (486 or later)
- 16 MB of RAM
- Windows 98/NT4/2000/XP, or WinPE 2.0
- 16-color VGA video or better

What's new in Symantec DeployCenter

Symantec DeployCenter has limited Vista support, as follows:

- Symantec DeployCenter Library tools run on WinPE 2.0. Symantec DeployCenter Library cannot be installed on Vista, and the tools will not run on Vista.
- Symantec DeployCenter Library tools can handle the disk layouts as seen by Vista disk manager.
- System restore points that are captured during image creation are removed upon image deployment.
- If a disk signature does not already exist on a target disk, Symantec DeployCenter Library tools create a new disk signature for the cloned disk and patch it into the registry of mounted devices in Vista.
- Symantec DeployCenter Library tools support new Vista boot manager configuration updates.

- Symantec DeployCenter Library tools work with non-cylinder aligned partitions. In restore operations an image file is restored with the same partition alignment as was captured.
- PQDisk can resize and move Vista partitions.
- When dealing with transactional NTFS, Symantec DeployCenter Library tools trigger the resource manager to reset itself to eliminate all unresolved transactions, if any, upon next boot.
- Symantec DeployCenter Library tools do not support BitLocker.
- A Virtual Floppy can be executed on Vista.

CHAPTER 1

PartitionInfo

Overview

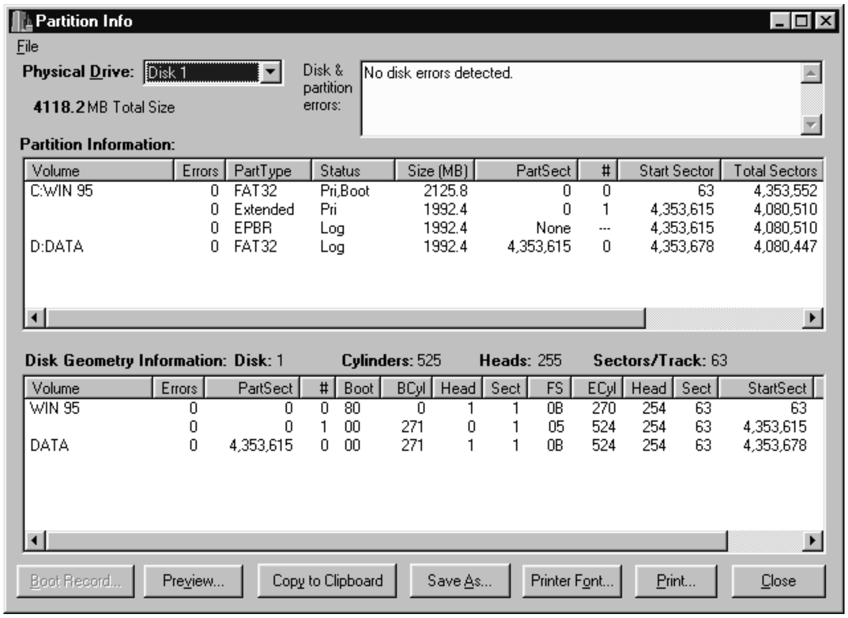
PartitionInfo is a Windows program that generates a report showing the contents of your hard disk partition table. This information is helpful in resolving various partitioning problems. You can run PartitionInfo under Windows 95, Windows 98, Windows Me, Windows NT 4.0 Workstation, and Windows 2000 Professional.

PartInfo is a DOS utility that provides essentially the same information without the GUI interface.

Running PartitionInfo

1 Click **Start ➤ Program Files ➤ Symantec ➤ PartitionInfo**.

The PartitionInfo window appears, displaying partition and disk geometry information and disk and partition errors. Disk geometry information includes data from the master boot record and the extended partition boot records.



Physical Drive: Disk 1

4118.2 MB Total Size

Disk & partition errors: No disk errors detected.

Partition Information:

Volume	Errors	PartType	Status	Size (MB)	PartSect	#	Start Sector	Total Sectors
C:WIN 95	0	FAT32	Pri,Boot	2125.8	0	0	63	4,353,552
	0	Extended	Pri	1992.4	0	1	4,353,615	4,080,510
	0	EPBR	Log	1992.4	None	---	4,353,615	4,080,510
D:DATA	0	FAT32	Log	1992.4	4,353,615	0	4,353,678	4,080,447

Disk Geometry Information: **Disk:** 1 **Cylinders:** 525 **Heads:** 255 **Sectors/Track:** 63

Volume	Errors	PartSect	#	Boot	BCyl	Head	Sect	FS	ECyl	Head	Sect	StartSect
WIN 95	0	0	0	80	0	1	1	0B	270	254	63	63
	0	0	1	00	271	0	1	05	524	254	63	4,353,615
DATA	0	4,353,615	0	00	271	1	1	0B	524	254	63	4,353,678

Only errors that display in the box near the top of the screen indicate problems. Do not be concerned with Warnings and Infos in the bottom two boxes.

2 From the **Physical Drive** drop-down list, select the disk for which you wish to view information.

3 You can save the PartitionInfo report as a file, or you can print it.

To do this:	Do this:
Save the report as a file	Click **Save As**. In the Filename box, type a name for the file. Click **Save**. (Columns of information are separated by tabs, so you can open the file in a word processor and easily format the report.)
To change the font for a printed report	Click **Printer Font**. Select the desired font, font style, size, and so forth. Click **OK**.
To change printer setup	Click **File ➤ Printer Setup**. We recommend that you set the page orientation to landscape to avoid text being cut off at the right margin.
To print a report	Click **Print**, then click **OK**.

4 To exit PartitionInfo, click **Close**.

Running PartInfo

1 Boot the computer to DOS.

2 You have several options for running PARTINFO.

To do this:	Do this:
To display partition information on your screen	Type `PARTINFO`, then press <Enter>.
To send a report directly to your printer	Type `PARTINFO >LPT1` or `PARTINFO >PRN`, then press <Enter>.
To save the report as a text file on a floppy disk	Type `PARTINFO >A:\PARTINFO.TXT`, then press <Enter>.

Debug Report

If you run PartInfo from DOS, you can generate a debug report. The debug report is used to produce output while the program is in operation. This output is used to troubleshoot various CHECK errors. The debug report contains elements of PartInfo, but is more comprehensive. Once the program errors out, the debug report gives you detailed, low-level partition information. This can narrow down where the error is occurring and possible causes for it in your deployment process.

To run PartInfo and generate a debug report, type the following at DOS:

```
PARTINFO /LVL=n
```

You can use the /LVL switch to specify different levels of information for the debug report. The level is specified by “`n`” in the example above. There are a total of seven levels. Level three is the default. For troubleshooting purposes, Symantec recommends using the Level 7 option.

CHAPTER 2

PQAccess

Overview

PQAccess is a scriptable command-line utility for accessing individual files and complete directories on partitions not normally accessible to the active operating system. For example, DOS cannot normally access NTFS partitions and some versions of DOS cannot access FAT32 partitions, but PQAccess allows files to be copied to and from these partitions while running in DOS. PQAccess also enables getting directory listings, creating directories, and deleting files and directories on partitions not normally accessible to the active operating system.

PQAccess can save money by reducing the time required to set up and configure new workstations, deploy software applications across a network, and upgrade systems to new operating systems or larger hard disks.

There are three versions of PQAccess:

- **PQAccessNT** is a Win32 application that runs in a Windows DOS box within Windows NT 4.0, Windows 2000, Windows PE, or Windows XP.
- **PQAccess9x** is a Win32 application that runs in a Windows DOS box within Windows 9x or Windows Me.
- **PQAccD** only runs in DOS.

Unless otherwise noted, all references to or examples using one version of the application also apply to the other version. In this chapter, “PQAccess” is used to refer to any or all versions.

System Requirements

- IBM PC or compatible (Pentium or higher)
- 16 MB RAM
- Windows 95, Windows 98, Windows Me, Windows 2000, Windows XP, Windows PE, or DOS

Running PQAccess

To run PQAccess, specify the program name followed by optional arguments. For example:

```
PQAccD {argument, [argument]…}          (DOS command prompt)
PQAccess9x {argument, [argument]…}       (Windows 9x DOS box)
PQAccessNT {argument, [argument]…}      (Windows NT DOS box)
```

You can run PQAccess in two modes: command line mode or script mode. In command line mode, you specify all commands on a single command line. In script mode, you specify the commands in a script file, and the command line specifies just the script filename. Using script mode overcomes the limitations in the length of the command line and allows you to specify more complicated operations.

If an error occurs while running, PQAccD will exit with an exit code of one (1). If an error occurs while running, PQAccess9x or PQAccessNT, they will exit with an exit code that is the same as the error number. If no error occurs, all will exit with a value of zero.

Command Line Syntax

When used on the command line, commands are preceded by either a forward slash (/) or a minus sign (-).

You can shorten commands, using any of the abbreviations listed under "Command Aliases" on page 26. Commands are not case-sensitive, and the embedded minus signs (-) can be replaced with underscores (_) or omitted completely. For example, the following are equivalent commands:

```
log-file
LogFile
LOG_FILE
log
```

On the command line, program commands may be followed by an equal sign and an argument to specify a value for the command. The argument may also be proceeded by a space instead of an equal sign, but in this case, the argument must not start with a "/" or "-" or it will be interpreted as the next command instead of as an argument to the previous command. For example, the following commands are all valid:

```
PQAccD /Silent=yes
PQAccD /Silent = yes
PQAccD /Silent yes
```

Spaces between arguments will be replaced with a single space. If you are attempting to pass a filename that contains multiple spaces, you must place the name in quotes so that the spaces are not eliminated.

You can specify a drive using a drive letter, a PQAccess drive number, or a volume label. If multiple partitions have the same volume label, the first partition encountered with the specified label will be affected. You can use the /INFO command to display the partition order and volume labels.

Script File Syntax

To run PQAccess in script mode, specify a single command to load a script file, then specify the remaining commands from within the script file. A PQAccess script file can have any extension, but “.pqc” is the recommended extension.

Script files are text files that have one command on each line. Blank lines and lines that start with a “;” or a “#” or a “//” are ignored and can be used for comments within the script file. When used in script files, commands are not preceded by a forward slash or a minus sign as they are on the command line.

Spaces are allowed before and after the command, and on either side of the equal sign. The equal sign is optional. For example, in a script file, the following commands are equivalent:

```
Silent=yes
    Silent    =     yes
Silent yes
```

Accessing Hidden Partitions

PQAccess assigns a partition number to every partition on all hard drives on the system. Free space on the drives may also be assigned a number. These numbers can be used in place of drive letters to access partitions. Use the /info command to get a listing of all partitions along with their numbers and volume labels.

The following is an example of the information that would display if you ran `PQAccess /info`.

```
Partition Information
Disk Part # Letter PqAccess Size(MB) Used(MB) Type        Label
==== ====== ====== ======== ======== ======== =========== ===========
  1     1    C:          1:     2502     1209 FAT32       WIN98
  1          *:          2:     5554     5554 Extended
  1     2    *:          3:     5554     3531 NTFS        Win2K
  2     1    *:          4:        8        2 Free
  2          *:          5:    17124    16865 Extended
  2     2    D:          6:     2039     1924 FAT32       SOURCE
  2     3    E:          7:     4040     1755 FAT32       PQI DATA
  2     4    *:          8:     6001     4651 NTFS        NTFS-2
  2     5    F:          9:     1412       55 FAT         FATTEST
  2     6    G:         10:     3067     2566 FAT32       FAT32BACKUP
  2     7    *:         11:      204        3 NTFS        NTFS-Hidden
  2     8    *:         12:      102       28 FAT32       FAT32-HDN
```

```
    2     9     *:       13:         259        0 Free
    2    10     *:       14:        2432        0 Free
```

This example shows two hard drives with a FAT32 and an NTFS partition on the first drive and a collection of partitions on the second hard drive. Running under DOS, only the FAT and FAT32 partitions are assigned drive letters. The third column lists the assigned drive letter if there is one. The fourth column lists the drive number that can be used by PQAccess in place of the drive letter to access FAT, FAT32, and NTFS partitions. Refer to the description of the /info command on page 26 for more information.

To copy a file from a hidden NTFS partition to a FAT32 partition on the first drive, you could use the following command:

```
PQAccess -copy 3:\docs\report.doc C:\mydocs\report.doc
```

The “C:” part of the destination could be replaced with “1:” or “:volumelabel:” and the program would behave identically.

PQAccess Commands

This section includes information about the commands supported by PQAccess.

Command Parameters

Command parameters are specified by placing a space or an equal sign (=) after the command, followed by a parameter value. The following table describes the available command parameters. For information about PQAccess commands, see “List of Commands” on page 14.

Parameter	Description
<file>	Indicates a filename, optionally including a drive letter and an absolute or relative path to the file. If a path is specified, then all directories in the path must already exist. The file is always accessed using the operating system’s I/O, so the name cannot include a drive number in place of a drive letter. The command *SCRIPT-FILE* requires that the file already exist and will cause an error if the specified file cannot be found. The commands *LOG-FILE* and *ERROR-FILE* will create new files with the specified name.
<afile>	Indicates a filename that must exist. It must include either a drive letter or a PQAccess drive number.

Parameter **Description**

<wfile>

<wfile> can refer to a single file or can include the wildcard characters “*” and “?” as part of the filename to refer to a group files matching the pattern in the specified directory. It must include either a drive letter or a PQAccess partition number. Characters entered after a wildcard are ignored. The directory portion must not include wildcard characters.

<src-file>
<dest-file>

<src-file> is a <wfile>. <src-file> and <dest-file> must be separated by either a space or a comma. If <src-file> includes embedded spaces, then it should be surrounded by quotes. For example, the following is an incorrect usage within a script file because <src-file> will be assigned “C:\Program” while <dest-file> will be assigned “Files\PQ\PQAcc.exe D:\progs\PQAcc.exe.”

```
copy C:\Program Files\PQ\PQAcc.exe D:\progs\PQAcc.exe
```

The correct way to specify this line is:

```
copy "C:\Program Files\PQ\PQAcc.exe" D:\progs\PQAcc.exe
```

<dest-file> must include a drive letter, drive number, or volume label and cannot include wildcard characters. <dest-file> does not need to be in quotes even if it contains embedded spaces, but it is permissible to quote it.

<path>

An absolute path specifying a local or network directory. It must include a drive letter, a PQAccess partition number, or a volume label. Syntax:

```
c:\windows
1:\windows
:win98:\windows
```

<yes/no>

For commands that take a <yes/no> parameter, the parameter is optional. If not specified, then yes is assumed. Several synonyms are also available. The strings “yes” “y” “on” “true” “t” and “enable” are all equivalent and enable the specified command mode. The strings “no” “n” “off” “false” “f” and “disable” all disable the specified command mode. These strings are not case-sensitive.

<text>

Any text string. If you want to have spaces or tabs before or after the printable characters, place the whole string inside of either single (') or double (“) quotes. Otherwise all spaces are stripped from the ends of the string.

List of Commands

For a list of abbreviations, see “Command Aliases” on page 26. For details about parameters for these commands, see “Command Parameters” on page 12.

Command	Parameter	Description
?	[<n>]	Displays a brief summary of all enabled commands. All of the commands in this list that have not been disabled or hidden in the RTC file will be displayed. This command can take a number as an optional parameter <n>. If specified and the help is being displayed to a window (that is, SILENT has not been specified), then the output will pause after each <n> lines of output are displayed. Use a value of zero for <n> to disable pausing. Default value for <n> is 22.
ABT	<yes/no>	Allows <Ctrl-C> to be used to terminate the program. When enabled, the program will terminate shortly after <Ctrl-C> is pressed. When disabled, the program will ignore <Ctrl-C>. Enabled by default.
ACCESS	<yes/no>	*DIR* option. Shows the last access date/time of each file listed. This data may not always be available on FAT partitions; FAT does not save the access time, only the date. Disabled by default.
AFX	<yes/no>	Automatically fixes partition errors, such as the CHS values not matching the LBA values, etc. Disabled by default.
ATTRIB	<attr> <wfile>	Change the attributes of <wfile>. <wfile> can include wildcards. The attributes that can be changed are: • `R = Read-Only` • `H = Hidden` • `S = System` • `A = Archived` To add an attribute specify a plus, “+”, before the letter. To remove an attribute specify a minus, “-“, before the letter. Multiple attribute changes can be specified at the same time. Any attribute not specified will not be changed. For example: `/att +R-SH c:\*.xyz` will add the read-only attribute to each file, and remove the system and hidden attributes if present. The archive attribute will remain unchanged for each file.

Command	Parameter	Description

ATTRIB
(cont'd)
		Note that when used on the command line, the minus sign cannot appear immediately after a space or it will be interpreted as the next command and not as an argument to the *attributes* command. For example,
		`/att -R+SH C:\*.xyz`
		would be invalid. Instead, use one of the following:
		`/att +SH-R C:\*.xyz`
		`/att=-R+SH C:\*.xyz`
		There is no similar problem when specifying attributes within a script file because in a script file, all commands are on a separate line and all text is treated as part of the argument to the command.

CHK	[fix]<path>	Performs a Symantec check-disk operation on the specified hard disk partition. This operation never uses the OS check-disk. The Symantec check-disk may not find some of errors that the OS check-disk would find, but it can detect a large number of partition errors on FAT, FAT32, NTFS, Ext2, and Ext3 partitions. You should specify the partition parameter using drive letter/drive number syntax with no path (such as /chkdsk c: or /chkdsk 3:). The check-disk function can also fix/repair some types of errors. It will only fix these errors if the string "fix" is specified before the partition. For example: `/chkdsk fix 3:`
CMD	<file>	Reads commands from a script file. The script file must exist. All commands in the script file will be executed before any more commands from the command line are processed. Refer to the section "Script Syntax" above, for information about the format of commands in a script file.

Command	**Parameter**	**Description**
COPY	<src-file><dest-file>	Copy a file or files from <src-file> to <dest-file>. If <src-file> is a single file then <dest-file> can be a file or directory. If <src-file> contains the wildcard characters "*" (match any number of characters) or "?" (match a single character) as part of the filename then <dest-file> must specify a directory, which must already exist. Both <src-file> and <dest-file> should include the full path to the file, including a drive letter, PQAccess drive number, or volume label. If you use wildcard characters in <src-file>, characters entered after the wildcard will be ignored. See the section titled "Accessing Hidden Partitions" for more details about using PQAccess drive numbers. See also the commands *SUB* and *USE-OS* for details on how these commands affect the *COPY* command. As each file is copied, it will be listed in the log and/or to the display, according to the settings of the LOG and SILENT commands. When a file is copied, the original file attributes and file times are also copied. This includes file creation, last modification and last access times/dates for each file that it restores. The attributes that will be restored are: hidden, system, read-only and archived. Note that NTFS supports additional attributes that are not copied. Note also that only the main stream of multi-stream NTFS files can be copied.
CRC	<yes/no>	*DIR* option. Compute a CRC for each file that is listed. Computing the CRC requires that the file be read from disk, so this option makes listing much slower. Disabled by default.
CREATE	<yes/no>	*DIR* option. Show the creation date/time of each file listed. This data may not always be available on FAT partitions. Disabled by default.
DEL	<wfile>	Delete a file or group of files. <wfile> can include wildcards. Characters entered after a wildcard are ignored. The *DELETE* command is never recursively applied to subdirectories.As each file is deleted, it will be listed in the log and/or to the display, according to the settings of the LOG and SILENT commands.

Command	Parameter	Description
DIR	<wfile>	List all of the files and directories contained specified by <wfile>. Include the size of each file. If the *SUB* option is set to *true*, then also list the files in each subdirectory. The commands *CRC, SHORT, SAT, MODIFY, CREATE* and *ACCESS* can be used to change the information shown during a listing.
DTR	<path>	Delete a directory and all of the files and subdirectories contained within this directory. The names of individual files and directories that are deleted using this command are not logged.
ECHO	<text>	Echoes the specified text to the screen and/or log file. If *silent* mode is not enabled, then the text will be displayed on the screen. If logging is enabled, it will also be written to the log file. You can use this command to display status information to the end user.
ERR	<file>	Records exit status in a file. Upon completion of the application, the error file will only exist if the application encountered an error. If a file with this name existed before the application ran, it will be deleted. The error file is a text file. The first line of the error file contains the error code (number). The next line in the file contains a description of the error. If multiple errors occur, only the first error will be recorded in this file. All errors will be written to the log file, if it is enabled.
EXIT		Exit immediately. This command is usually not needed. It exists so that if all commands are specified within the RTC file, PQAccess can be commanded to exit immediately after completing the RTC commands, without processing any commands from the command line or from a script file.
IDE	<yes/no>	*(DOS only)* When accessing an IDE hard disk using PQ direct methods as opposed to OS calls, the disk is normally accessed using BIOS interrupt 13. This switch uses UDMA accesses instead, which is generally much faster. However, UDMA access does not always work, especially on older systems. This command is only available in the DOS version. It is disabled by default.

Command	Parameter	Description
INFO		Shows information about each partition and free space on all system hard drives. The first two columns display the disk number and partition number as used in Symantec® Drive Image® scripting. The third column specifies the drive letter assigned to the partition by the current operating system, if applicable. The fourth column specifies the PQAccess number you can use to refer to a partition instead of the drive letter. The PQAccess number (including the colon) can be specified to PQAccess as part of the *DEST-PATH*, in place of the drive letter. Use the PQAccess number so PQAccess can use partitions not visible to the operating system. The remaining columns display the partition size, the amount of used space within the partition, its type and its label. This is the same information displayed when using the /INFO switch with Symantec Drive Image.
LOG	<file>	Logs information to file. Log information about the status of all commands to a file. All error information will be recorded in the log file. If a file with this name existed before running, it will be overwritten. The log-file should generally be the first command on the command line or in the script
MKDIR	<path>	Create a new directory. This command will display a warning if the directory already exists, but not generate an error.
MODIFY	<yes/no>	*DIR* option. Show the last modification date (change date) and time of each file listed. Enabled by default.

Command	Parameter	Description
MOVE	<src-file><dest-file>	Move a file or files from <src-file> to <dest-file>. If <src-file> is a single file then <dest-file> can be a file or directory. If <src-file> contains the wildcard characters "*" (match any number of characters) or "?" (match a single character) as part of the filename then <dest-file> must specify a directory, which must already exist. Both <src-file> and <dest-file> should include the full path to the file, including a drive letter or PQAccess drive number. Characters entered after a wildcard are ignored. See the section titled "Accessing Hidden Partitions" for more details about using PQAccess drive numbers. See also the commands *SUB* and *USE-OS* for details on how these commands affect the *MOVE* command. If <src-file> and <dest-file> are on separate partitions, this command will copy each file and then delete the file from its original location. If <src-file> and <dest-file> are on the same partition, then this command behaves similar to *REN*, except that <src-file> can include wildcards. In this case, each file will have its directory entry updated to reflect the move, but the data in each file does not physically need to be moved, so moving is much faster. As each file is moved, it will be listed in the log and/or to the display, according to the settings of the LOG and SILENT commands.
NOX		Clear all of the exclude lists created by exclude-dir, exclude-file, exclude-wildcard or exclude-attrib, so that all files are again processed.
OVW	<yes/no/always>	If you copy a file and a file with the same name already exists in the destination location, this command determines the behavior of the *copy*. If *OVW* is set to "yes", then the previously existing file will be overwritten. If *OVW* is set to "no", then an error will occur. If *OVW* is set to "always" or "replace" or "a" or "r", then the copy will only succeed if the file already exists in the destination location. It will fail if there is no file to overwrite. The "always" option will probably rarely be used. The default is "yes."

Command	Parameter	Description
REN	<afile><new-name>	Rename a file and/or move it to a different location on the same partition. This command does not support using wildcards as part of <afile>. <new-name> does not need to specify a drive letter/number, but if it does, it must be the same as the drive specified to <afile>. If <new-name> consists only of a filename with no path information, then <afile> will be renamed to the name specified by <new-name>. Otherwise <afile> will be moved from its current location to the location specified by <new-name>. The location must also include the new name for the file (even if the name is staying the same).
RMDIR	<path>	Delete a directory. The directory must be empty or an error will occur. It is also an error if the directory does not exist when this is called.

SAT	<yes/no>	*DIR* option. Show the file attributes of each file listed. Enabled by default.
		For FAT, FAT32 and NTFS partitions, the attributes are:
		R = Read-Only
		H = Hidden
		S = System
		A = Archived
		D = Directory
		A file may have multiple attributes.

SHORT	<yes/no>	*Dir* option. Show the short (8.3) name of each file, but only if the short name is different from the long name. Note that Ext2 and Ext3 file systems do not have separate short and long filenames, and NTFS partitions can be configured to not have separate short names. Disabled by default.
SILENT	<yes/no>	Silent mode. If *silent* mode is enabled, most text messages are not displayed on the screen. Progress and error messages will continue to be displayed. By default, silent mode is disabled.

Command	Parameter	Description
SUB	<yes/no>	Recursively processes subdirectories. If the *COPY* command or *MOVE* command is called using wildcards, then the wildcard mask will be applied to all files in the specified directory and all subdirectories of the specified directory. By default, *COPY* and *MOVE* only copy files from the specified directory.
TCA	<time> <wfile>	Similar to the *TCM* command, except that *TCA* changes the last accessed time of <wfile>.
TCC	<time> <wfile>	Similar to the *TCM* command, except that *TCC* changes the creation time of <wfile>.
TCM	<time> <wfile>	Change the last modified time of <wfile>. <wfile> can include wildcards. The last modified time is the file time that people are most familiar with. <time> is of the form: [[YY]YY]MMDDhhmm[.ss]. MM is month (01-12: requires that you use a leading 0 if month is a single digit); DD is day of month (01-31); hh is hour (00-23); mm is minutes (00-59); ss is seconds (00-59); YY is two digit year (70-99, 00-69 representing 1970-2069); YYYY is four digit year (1970-2099, years outside of this range are not supported). You can also specify the word “now” or omit the <time> parameter to use the current time.

Command	Parameter	Description
USE-OS	<yes/no/src/dest>	If the partition or partitions specified are visible to the OS (generally this means that it has been assigned a drive letter), this commands instructs the application to use OS calls to perform the commands specified in the PQA file. When accessing a hidden partition, the setting of this command is ignored, because PQAccess cannot use OS calls to access the partition. Unless the partition has recently been modified without the operating system's knowledge (perhaps by a program such as Drive Image or Norton PartitionMagic), you should leave this command enabled. **Use great care if you disable this command.** If the operating system has not flushed all of its caches before PQAccess runs, or if the OS attempts to access the target partition after PqaDeploy has modified it without first rebooting, the partition may become corrupted. OS access is also generally faster than non-OS access. For commands such as *COPY* and *MOVE*, it is possible to work with two partitions. You can set *USE-OS* to "source" or "src" or "s" to instruct PQAccess to use OS calls when reading the source files but to not use OS calls when writing to the destination partition. You can set *USE-OS* to "destination" or "dest" or "dst" or "d" to instruct PQAccess to not use OS calls when reading the source files but to use OS calls when writing to the destination partition. When you use commands that don't work with two partitions, such as *delete, mkdir* or *list*, the setting specified for the "destination" will be used. By default, this command is enabled (for both source and destination).
VER		Shows version information. If logging is enabled, this information will be recorded in the log. This information will display to the screen only if *SILENT* mode is not enabled.

Command	Parameter	Description

XAT	<attribs>	Exclude files that have or don't have the specified attributes. Directories attributes are not checked. Possible attributes include the following:
		R = Read-Only
		H = Hidden
		S = System
		A = Archived
		C = Compressed
		N = Normal
		N is a special case. When used, it must be the only attribute specified and causes all files to pass the attribute test (default). N is useful for clearing a previous attribute, without using the no-exclude command. Specifying a minus before an attribute means exclude any file that does not have that attribute. The minus affects all following specified attributes, until a plus sign is seen. Specifying a plus or not specifying any sign before an attribute means to exclude any file that has that attribute. For example, `/XAT HS-R` excludes from being processed, any file that is a hidden file or a system file or any file that is not read-only. The following commands can have files excluded during processing, but only when using wildcards with the command: *DIR, COPY*, *MOVE*, *TCM*, *TCA*, *TCC*, *ATTRIB*, and *DEL*.

XDR	<dirs>	Exclude the specified directory or directories from being traversed during recursive calls for the commands listed below. The calls are cumulative, so that if *XDR* is called multiple times each directory is added to the list of excluded directories. It is also possible to specify multiple directories at the same time, by separating each directory with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the directory name. The list of excluded directories can only be cleared by a call to *NOX.*When specifying a directory name, the absolute path should be specified, starting with the \ of the root directory. The drive letter/number followed by a colon should not be included before the path.

Command	Parameter	Description
		For example, the command: `PQAccess -rec -xdr \windows\system32 -copy c:\windows\* d:\winbak` copies all files and directories in the C:\windows directory, except those in the \windows\system32 directory. The following commands can have directories excluded during recursive processing: *DIR*, *COPY*, and *MOVE*. With the *COPY* and *MOVE* commands, the source directory is the one checked for a match with the excluded directories
XFL	<files>	Exclude the specified file or files from being used during processing of any of the commands listed below. The calls are cumulative, so that if *XFL* is called multiple times each file is added to the list of excluded files. It is also possible to specify multiple files at the same time, by separating each filename with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the filename. The filename must include the full path of the file, and cannot include any wildcard characters. The list of excluded files can only be cleared by a call to *NOX*. When specifying a filename, the absolute path should be specified, starting with the \ of the root directory. The drive letter/number followed by a colon should not be included before the path. The following commands can have files excluded during processing, but only when using wildcards with the command: *DIR,* *COPY*, *MOVE*, *TCM*, *TCA*, *TCC*, *ATTRIB*, and *DEL*. With the *COPY* and *MOVE* commands, the source filename is the one checked for a match with the excluded filenames.

Command	Parameter	Description
XWC	<masks>	Exclude any files whose filename matches any of the specified wildcard masks from being used during processing of any of the commands listed below. The calls are cumulative, so that if *XWC* is called multiple times each mask is added to the list of masks. It is also possible to specify multiple masks at the same time, by separating each wildcard mask with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the mask. The masks cannot include directory information, as they are only applied to the filename portion of each file, and not to the full path name. The list of excluded masks can only be cleared by a call to *NOX*. The following commands can have files excluded during processing, but only when using wildcards with the command: *DIR, COPY*, *MOVE*, *TCM*, *TCA*, *TCC*, *ATTRIB*, and *DEL*.

Command Aliases

You can substitute commands to use any of the following aliases.

Command	Alias
?	h help
ABT	allow-abort allow-ctrl-c
ACCESS	acc access-date
AFX	auto-fix
ATTRIB	att attributes
CHK	chk-dsk check-disk
CMD	script script-file
COPY	cp cpy
CRC	show-crc compute-crc
CREATE	cre create-date
DEL	delete remove
DIR	lst list
DTR	del-tree delete-tree
ECHO	prn print
ERR	error error-file
EXIT	die terminate
IDE	udma enable-udma
INFO	inf partition-info
LOG	log log-file
MKDIR	md make-dir
MODIFY	mod modify-date
MOVE	mv mov
NOX	xno no-x x-none no-exclude exclude-none
OVW	replace over-write
REN	rn rename
RMDIR	rd del-dir
SAT	show-attr show-attrib
SHORT	snm short-name
SILENT	sil quiet
SUB	rec recursive
TCA	touch-a touch-access
TCC	touch-c touch-create
TCM	touch-m touch
USE-OS	uos use-os-calls
VER	version
XAT	x-attr x-attrib exclude-attr exclude-attrib
XDR	x-dir exclude-dir
XFL	x-file exclude-file
XWC	x-wc x-wildcard exclude-wc exclude-wildcard

Script File Example

When PQAccess is only used to copy a single file or perform a simple command, then all the commands can be specified on the command line (as in the example under “Accessing Hidden Partitions” on page 11). However, if several operations need to be performed, it is easier to place these into a script file as listed in the example below:

```
// GetCAB.pqc
// Restore all of the CAB files from a utility
// partition visible as drive C: to an NTFS
// partition not currently visible to the DOS.
// so that Windows XP can be re-installed.

// Display a status message to the user

echo Restoring Windows files. This may take some time.

// Disable most messages to the screen
Silent

// Record Results to a file in case it is needed by
// tech support.
Log = c:\recover.log

// Add PQAccess version information to the log file.
Version

// Restore files
copy C:\WINNT\options\*.cab 2:\WINNT\options\

// End of file
```

To run the application, you would then use the command:

```
PQAccD /cmd=c:\getcab.pqc
```

RTC Files

A runtime configuration file (RTC) is supplied for each version of PQAccess. The RTC file has the same name as the corresponding executable, but the extension has been changed from .EXE to .RTC. Any PQAccess command or group of commands can be added to the RTC file, so that these commands are executed every time the application runs. Any PQAccess command can also be disabled in the RTC file, so that the command is not available to customers, preventing them from using PQAccess in ways not intended. It is also possible to hide commands so that they don't display when the help command is used, but so that they can still be called.

All versions of PQAccess also have the ability to “lock” themselves to run only on computers with a specific BIOS or DMI string. BIOS information is specified in the RTC file. When locking is enabled, the system BIOS or DMI is checked before PQAccess is allowed to execute. If the lock test fails, a customizable error message is displayed indicating the failure.

CHAPTER 3

PQQuery

Overview

PQQuery is a DOS/Windows command line program that returns disk and partition information in environment variables. The DOS version sets the values of the environment variable directly (unless the /BAT command line switch is used). The Windows version cannot set the environment variables directly, so the /BAT command line switch must be used. PQQuery32 can be used any place that PQQuery is used, unless otherwise noted.

PQQuery allows a batch file to test environment variables and perform different actions depending on the returned values.

If there is an error when creating the environment variables, all the variables previously created are deleted to avoid cluttering the environment. The command line arguments and environment variables created are defined below.

Calling Syntax

The calling syntax for PQQUERY is as follows:

```
PQQUERY[32] [/?] [/BAT=<file>] [/KB|MB|GB] [/V] [/DSK=(n|*)
[/PRT=(m|*)]] [/DP=x]
```

The /? Switch will cause the application to print a brief description of the application, including a description of all of the arguments and then exit immediately.

The /DP switch can be used to specify the decimal precision of the sizes. Set it to the number of decimal places after the decimal point that you would like. The default is 1. The maximum value is 3 if the /KB switch is used, otherwise the maximum value is 6.

If no arguments are given, the only environment variable created is DISKS, the number of disks installed in the system.

The “/DSK” command line argument specifies a disk number <n> or all disks, “*”. The “/PRT” command line argument specifies a partition number <m> or all partitions, “*”. The value specified may be a number from 1 to 99, or it may be a ‘*’ which indicates all drives or all partitions. When both “/DSK” and “/PRT” are specified, n may be ‘*’ for both /DSK and /PRT, or it can be a number for /DSK and ‘*’ for /PRT, but it cannot be a ‘*’ for /DSK and a number for /PRT. Both /DSK and /PRT may be specified in either lower or upper case. If /KB is specified, it causes all values to be specified in kilo-bytes. If /GB is specified, it causes all values to be displayed in gigabytes. If neither /KB nor /GB is specified, then values are specified in mega-bytes. The /MB switch may be specified for clarity to indicate megabytes, but since that is the default, it does not change the application behavior.

If there is an error encountered when creating the environment variables, all of the variables created will be deleted to avoid cluttering the environment. For example, if the disk or partition number specified does not exist then the variables that have been created are deleted and an error message is given.

If there is an error when starting the engine, no environment variables are created and the exit code is returned as the DOS ERRORLEVEL. See “Error Codes” on page 35 for a full list of the possible error codes.

Examples of valid arguments (assuming the disks and partitions exist):

```
PQQuery /dsk=1

PQQuery /dsk=*

PQQuery

PQQuery /dsk=1 /prt=*

PQQuery /dsk=2 /prt =1
```

Examples of invalid arguments:

```
PQQuery /dsk=* /prt=2

PQQuery /prt=*

PQQuery /prt=1

PQQuery /dsk=0

PQQuery /prt=1 /dsk=1
```

Environment Variables

Environment variables are created based on the specified command line arguments. In the table below, the “D” in variable names refers to disk and the “P” refers to partition. “n” is a number ranging from 1 to 99.

Variable Type	Variable Name	Definition	Possible Values
Number of disks	DISKS	The number of disks in the computer.	Integer >= 1 and < 100

Variable Type	Variable Name	Definition	Possible Values
Disk	DnPRTS	The total number of partitions (excluding extended partitions).	Integer >= 1 and < 100
	DnSZ	The total disk size of the disk.	Floating point value >= 0
	DnASZ	The total disk space that has been allocated to partitions on the disk (excluding extended partitions).	Floating point value >= 0
	DnUASZ	The amount of the space on the disk that has not been allocated to any partition.	Floating point value >= 0
Partition	DnPmSZ	The size of the partition.	Floating point value >= 0
	DnPmUSZ	The amount of the partition that is used. See “Note” on page 32.	Floating point value >= 0
	DnPmUNSZ	The amount of the partition that is not used. See “Note” on page 32.	Floating point value >= 0
	DnPmTYP	The type of the partition.	{FAT, FAT32, NTFS, HPFS, EXT2, EXT3, LinuxSwap, Unformatted, Free}
	DnPmSTS	The status of the partition.	{ACTIVE, HIDDEN, NONE}
	DnPmCLS	The class of the partition.	{PRIMARY, LOGICAL}
	DnLUNSZ	The number <m> of the partition with the largest amount of unused space (largest DnPmUNSZ value). This variable is only set if /PRT=* and some partition has unused space.	Integer >= 1 and < 100

Variable Type	Variable Name	Definition	Possible Values
	DWLUNSZ	Set to the number of the disk <n> that contains the partition that has the largest amount of unused space. Only set if /DSK=* and /PRT=*.	Integer >= 1 and < 100
	PWLUNSZ	Set to the number of the partition <m> on the disk specified by DWLUNSZ that has the largest amount of unused space. Only set if / DSK =* and /PRT=*.	Integer >= 1 and < 100
	SZLUNSZ	Set to the size of the unused space of the partition specified by DWLUNSZ and PWLUNSZ variables, i.e. the partition with the most unused space. Only set if / DSK =* and /PRT=*.	Floating point value > 0

NOTE: The used and unused size is only calculated for partitions of type FAT/FAT32 and NTFS. For all other types the used size is the size of the partition, whether any of it is used or not, and the unused size is always zero.

Environment Variable Examples

```
D1SZ=10000431

D1P1TYP=FAT32

D2UASZ=36

D2P1STS=HIDDEN

DISKS=2

D1PRTS=4

D2P2USZ=8249346
```

If the environment size is not large enough to hold all of the environment variables, an error message is output and the variables already created are deleted. To avoid this problem specify the environment size when loading DOS. This is done in the CONFIG.SYS file with the SHELL command. For example:

```
   SHELL=A:\COMMAND.COM /e:1024
```

Here, the /e switch specifies the size of the environment in bytes. The correct path to COMMAND.COM should be given. For most versions of DOS, the default size is 256 bytes. The maximum size is 32,768 bytes. In most cases, an environment size of 1024 bytes should provide plenty of space to create the environment variables.

The /V option can be used to display a summary of all of the disks and partition on the system, displaying a table that contains all of the information that would be set in the environment if "/DSK=* /PRT=*" were specified. This is similar to disk info in some other applications (or PartitionInfo in Symantec applications). Note that for the DOS version of PqQuery, the environment variables are still set, even if /V is used. For the Windows version, you can use /V without /BAT if you simply want to see the disk information output by this command.

/BAT command line directive

The /BAT=<file> switch can be used to change the behavior of the application, so that instead of setting environment variables directly, it creates a batch file named <file> that contains all of the commands necessary to set the environment variables that otherwise would have been set by the application. You must use /BAT with PqQuery32. For example:

```
PQQuery
```

directly sets the environment variable DISK, while

```
PQQuery /BAT=q.bat
CALL q.bat
```

accomplishes the same thing, but PQQuery does not directly modify any environment variables. Instead q.bat consists of one line, "SET DISKS=2" that sets the environment variable when q.bat is run.

Scenario Examples

- If no arguments are given, only the DISKS environment variable is created specifying the number of disks installed in the system. For example:

Command Line	Variables Created
PQQuery	DISKS

- If just the "/DSK" argument is given, the DISKS variable is created along with the DnPRTS, DnSZ, DnASZ, DnUASZ variables specifying the number of partitions, the total disk size, the space used by the partitions and the unassigned disk space. If a '*' is specified then the four disk variables are created for each disk. For example (assume 2 disks):

Command Line	Variables Created
PQQuery /DSK=1	DISKS D1PRTS, D1SZ, D1ASZ, D1UASZ

Command Line	Variables Created
PQQuery /DSK=*	DISKS D1PRTS, D1SZ, D1ASZ, D1UASZ D2PRTS, D2SZ, D2ASZ, D2UASZ

- If both the "/DSK" and "/PRT" arguments are given, the DISK variable, the four disk variables, and the six partition variables are created. For example:

Command Line	Variables Created
PQQuery /DSK=2 /PRT=1	DISKS D2PRTS, D2SZ, D2ASZ, D2UASZ D2P1SZ, D2P1USZ, D2P1UNSZ, D2P1TYP, D2P1STS, D2P1CLS

- If a number is specified for /DSK and a '*' is specified for /PRT, then the DISKS variable, the four disk variables for the specified disk, and the six partition variables for all of the partitions on the disk are created. For example: DISKS

Command Line	Variables Created
PQQuery /DSK=2 /PRT=*	DISKS D2PRTS, D2SZ, D2ASZ, D2UASZ D2P1SZ, D2P1USZ, D2P1UNSZ, D2P1TYP, D2P1STS, D2P1CLS D2P2SZ, D2P2USZ, D2P2UNSZ, D2P2TYP, D2P2STS, D2P2CLS

- If a '*' is specified for both arguments, the DISKS variable is created, the four disk variables for all disks are created, and the six partition variables for every partition on every disk are created. For example:

Command Line	Variables Created
PQQuery /DSK=* /PRT=*	DISKS D1PRTS, D1SZ, D1ASZ, D1UASZ D2PRTS, D2SZ, D2ASZ, D2UASZ D1P1SZ, D1P1USZ, D1P1UNSZ, D1P1TYP, D1P1STS, D1P1CLS D2P1SZ, D2P1USZ, D2P1UNSZ, D2P1TYP, D2P1STS, D2P1CLS D2P2SZ, D2P2USZ, D2P2UNSZ, D2P2TYP, D2P2STS, D2P2CLS

Error Codes

PQQuery has both an error code and an exit code. The DOS ERRORLEVEL variable will be set to the exit code value. If no errors occur, both the error code and the exit code will have a value of zero. The table below describes the error codes possible in PQQuery.

When PQQuery has an error between 5001-5200, the exit code will be the error code minus 5000. For example, if PQQuery encounters an error 5003, ERR_OUT_MEMORY, the application exit code will be 3. Two additional exit codes are possible. Exit code 255 results from an invalid command line argument. Exit code 254 results if an error code is outside the range of 5001-5200.

Error	Name	Description
0	OK	
5001	ERR_OSNOTSUPPORTED	A system that does not use Win32 Windows, Win32 NT, DOS16 or DOS32.
5002	ERR_INITFAILED	
5003	ERR_OUT_MEMORY	Out of memory
5004	ERR_UNINITFAILED	
5005	ERR_NODRIVES	
5006	ERR_DISKREAD	
5007	ERR_DISKWRITE	
5008	ERR_DUPLICATEPARTITION	
5009	ERR_CYLINDER	Too many cylinders for the BIOS structure
5010	ERR_NO_DMA_BUFFER	No dma safe buffer could be created for DOS
5011	ERR_MBRSIGNATURE	MBR read with bad signature
5012	ERR_BAD_EPBR_CHAIN	An EPBR referenced backwards to another epbr
5013	ERR_MULTIPLE_EXTENDED	There was more than one extended partition in an MBR table

Error	Name	Description
5014	ERR_PARTITIONOVERLAP	Partitions overlap
5015	ERR_INCONSISTENT_LBA_CHS	Inconsistent CHS and LBA sector count
5016	ERR_PAST_END_OF_DRIVE	A partition table entry extends past the end of the drive
5017	ERR_NOT_ON_CYL_BOUNDARY	The partition table entry does not end on a cylinder boundary
5018	ERR_BOOT_SIGNATURE	The signature was not found in the boot sector of a partition
5019	ERR_BSSECTORCOUNT	The boot sector has a different sector count than the partition entry
5020	ERR_CANT_REBOOT	Error trying to shutdown (reboot)
5021	ERR_REBOOT_FAILED	
5022	ERR_DRIVE_LETTER_NOT_FOUND	PQQuery could not find a partition for the given drive letter
5023	ERR_NO_VOL_LABEL	There is no volume label to return
5024	ERR_REAL_MEMORY	Error allocating real mode memory
5025	ERR_DPMI	Error executing a DPMI function
5026	ERR_FRS_TABLE	An error was found in an FRS table
5027	ERR_INVALID_PART	An invalid partition was specified.
5028	ERR_INVALID_DISK	An invalid disk was specified.
5029	ERR_INIT_ENV	The environment was not initialized correctly.
5030	ERR_CREATING_VAR	An error occurred when creating an environment variable.

CHAPTER 4

PQIExtract

Overview

PQIExtract is a utility for extracting individual files and complete directories from a PQI file. The PQI file is an image or snapshot of one or more partitions on the user's hard drive. The PQI file is created using one of Symantec's imaging products (ImageCenter, Drive Image, Drive Image Pro, EasyRestore, PQImage, or DISE.)

If Drive Image is used to restore the complete image to a user's hard disk, any data that has changed on the hard disk since the image was created is lost. In particular, new user data files and modified settings are lost, as well as any new applications that have been installed. PQIExtract allows restoring of individual files from within a PQI file, without modifying other existing files on the hard disk.

For example, suppose a user accidentally deletes a critical Windows file from his drive, preventing Windows from booting. If the user knows which file he has deleted, he can restore this single file from the PQI, allowing Windows to boot again. More commonly, the user will not know which file or files have been deleted or corrupted. In this case, PQIExtract can be used to extract all the Windows CAB files. The user can then run the Windows setup program to reinstall Windows. This allows Windows to be restored without losing any user data on the drive, though some user settings will be lost.

There are three versions of PQIExtract:

- PQIEXTNT is a Win32 application that runs in a Windows DOS box within Windows NT 4.0, Windows 2000, Windows PE, or Windows XP.
- PQIEXT9X is a Win32 application that runs in a Windows DOS box within Windows 9x or ME. Both the 9X and NT can handle long filenames and can access network drives specified using "\\machine\mountdir\path" where "machine" is the machine's network name, and "mountdir" is the network mount point on the machine.
- PQIEXTD only runs in DOS. It can handle long filenames when restoring files to the hard drive, but will fail if you attempt to restore files with long filenames to a network (remote) drive.

Unless otherwise noted, all references to or examples using one version of the application also apply to the other versions. In this document, PQIExtract is also used to refer to any or all versions.

Running PQIExtract

To run PQIExtract, specify the program name followed by optional arguments. For example:

```
PQIEXTD {argument, [argument]...} (DOS command prompt)
PQIEXT9X {argument, [argument]...}(Windows 9x DOS box)
PQIEXTNT {argument, [argument]...}(Windows NT DOS box)
```

PQIExtract may be run in either of two modes: *command line* mode or *script* mode. In command line mode, all commands are specified on a single command line. In script mode, the commands are specified in a script file and the command line specifies just the script filename. Running in script mode overcomes the limitations in the length of the command line and allows more complex operations to be specified.

If an error occurs while running PQIEXTD, it will exit with an exit code of one (1). If an error occurs while running PQIEXT9X or PQIEXTNT, the programs will exit with an exit code that is the same as the error number. If no error occurs, all will exit with a value of zero.

Command Line Syntax

When used on the command line, commands are preceded by either a forward slash (/) or a minus sign (-). Commands can be shortened to use any of the abbreviations listed under “Command Aliases” on page 51. Commands are not case-sensitive, and the embedded minus signs (-) can be replaced with underscores (_) or omitted completely. For example, the following are equivalent commands:

```
log-file
LogFile
LOG_FILE
log
```

On the command line, program commands can be followed by an equal sign followed by an argument to pass a value to the command. The argument can also be proceeded by a space instead of an equal sign, but in this case, the argument must not start with a “/” or “-” or it will be interpreted as the next command, instead of as an argument to the previous command. For example, the following commands are all valid:

```
PQIExtNT /Silent=yes
PQIExtNT /Silent = yes
PQIExtNT /Silent yes
```

Spaces between arguments will be replaced with a single space. If you are attempting to pass a filename that contains multiple spaces, you must place the name in quotes so that the spaces are not eliminated.

You can specify a drive using a drive letter, a PQAccess drive number, or a volume label. If multiple partitions have the same volume label, the first partition encountered with the specified label will be affected. You can use the /INFO command to display the partition order and volume labels.

Script File Syntax

To run PQIExtract in script mode, you specify a single command to load a script file, and then specify the remaining commands from within the script file. A PQIExtract script file can have any extension, but .pqe is the recommended extension.

Script files are text files that have one command on each line. Blank lines, and lines that start with a “;” or a “#” or a “//” are ignored and can be used for comments within the script file. In script files, commands are not preceded by a forward slash or a minus sign as they are on the command line. Spaces are allowed before and after the command and on either side of the equal sign. The equal sign is optional. For example, in a script file, the following forms are equivalents:

```
Silent=yes
  Silent   =   yes
silent yes
```

Accessing Hidden Partitions

PQIExtract assigns a partition number to every partition on all hard drives on the system. Free space on the drives may also be assigned a number. These numbers can be used in place of drive letters to access partitions. Use the /info command to get a listing of all partitions along with their numbers.

The following is an example of the information that would display if you ran `PQIExtNT /info`.

```
Partition Information
Disk Part # Letter PqAccess Size(MB) Used(MB) Type        Label
==== ====== ====== ======== ======== ======== =========== ===========
  1     1     C:       1:       2502     1209 FAT32       WIN98
  1           *:       2:       5554     5554 Extended
  1     2     *:       3:       5554     3531 NTFS        Win2K
  2     1     *:       4:          8        2 Free
  2           *:       5:      17124    16865 Extended
  2     2     D:       6:       2039     1924 FAT32       SOURCE
  2     3     E:       7:       4040     1755 FAT32       PQI DATA
  2     4     *:       8:       6001     4651 NTFS        NTFS-2
  2     5     F:       9:       1412       55 FAT         FATTEST
  2     6     G:      10:       3067     2566 FAT32       FAT32BACKUP
  2     7     *:      11:        204        3 NTFS        NTFS-Hidden
  2     8     *:      12:        102       28 FAT32       FAT32-HDN
  2     9     *:      13:        259        0 Free
  2    10     *:      14:       2432        0 Free
```

This example shows two hard drives with a FAT32 and an NTFS partition on the first drive and a collection of partitions on the second hard drive. Running under DOS, only the FAT and FAT32 partitions are assigned drive letters. The third column lists the assigned drive letter if there is one. The fourth column lists the drive number that can be used by PQAccess in place of the drive letter to access FAT, FAT32, and NTFS partitions. The drive numbers can also be used by PQIExtract to access the corresponding partition. Refer to the description of the /info command on page 44 for more information.

To extract a group of files to a hidden NTFS partition, you could use the following command:

```
PQIExtract -pqi=image.pqi -src=\appdir\* -dest=3:\appdir -copy
```

Command Parameters

Command parameters are specified by placing a space or an equal sign, "=", after the command, followed by a parameter value. The following table describes the available command parameters. For information about PQIExtract commands, see "PQIExtract Commands" on page 41.

Parameter	Description
<file>	Indicates a filename, optionally including a drive letter and an absolute or relative path to the file. If a path is specified, then all directories in the path must already exist. The commands *PQI* and *CMD* require that the file already exist, and will cause an error if the specified file cannot be found. The commands *LOG* and *ERR* will create new files with the specified name.
<path/file>	Allows you to specify the full path to either a directory or a file. You can specify a path using a drive letter, drive number, or volume label.
<yes/no>	For commands that take a <yes/no> parameter, the parameter is optional. If not specified, then *yes* is assumed. Several synonyms are also available. The strings "yes" "y" "on" "true" "t" and "enable" are all equivalent and enable the specified command mode. The strings "no" "n" "off" "false" "f" and "disable" all disable the specified command mode. These strings are not case sensitive.
<password>	A valid password string. This string is unencrypted, so your password may be revealed if the customer can see this command.

Parameter	Description
<path>	An absolute or relative path specifying a directory. For the command *DST*, the path can include a drive letter or volume label (for example, c:\windows, 1:\windows, and :win98:\windows are equivalent). Any directory in the path that does not exist will be created. For the *SRC* command, the path must be an absolute path that exists within the PQI file. If the PQI file contains multiple partitions, you should begin the path with `\partition#\`, followed by the path within the partition (where `#` is the partition number, with the first partition having a number of zero). If xx is omitted, the path will be treated as if the first partition was specified, so `\partition0\windows` and `\windows` would be equivalent.
<text>	Any text string. If you want to have spaces or tabs before or after the printable characters then place the whole string inside of either single (‘) or double (“) quotes. Otherwise all spaces are stripped off from the ends of the string.

PQIExtract Commands

Several commands are required to do anything useful with PQIExtract. Since PQIExtract is used to *COPY*, *CMP*, *DIR* or *RESTORE* files, one or more of these commands are required for the program to provide any functionality. These commands in turn must be preceded by the commands *PQI*, *SRC* and *DST*. (*DST* is not required for the *DIR* command.) A minimal script will contain four commands.

The supported commands are listed in the following table. For additional details about parameters for these commands, see “Command Parameters” on page 40.

Command	Parameter	Description
?	[<n>]	Displays a brief summary of all enabled commands. All of the commands in this list that have not been disabled or hidden in the RTC file will be displayed. This command can take a number as an optional parameter <n>. If specified and the help is being displayed to a window (i.e. silent has not been specified), then the output will pause after each <n> lines of output are displayed.Use a value of zero for <n> to disable pausing. Default value for <n> is 22.
ABT	<yes/no>	Allows <Ctrl-C> to be used to terminate the program. When enabled, the program will terminate shortly after <Ctrl-C> is pressed. When disabled, the program will ignore <Ctrl-C>. Enabled by default.

Command	Parameter	Description
ACCESS	<yes/no>	*DIR* option. Show the last access date/time of each file listed. This data may not always be available on FAT partitions; FAT does not save the access time, only the date.
AFX	<yes/no>	Automatically fixes partition errors, such as the CHS values not matching the LBA values, etc. Disabled by default.
CMD	<file>	Reads commands from a script file. The script file must exist. All commands in the script file will be executed before any more commands from the command line are processed. Refer to the section "Script File Syntax" above, for information about the format of commands in a script file.
CMP		Compares files in the PQI file as specified by *SRC* with those on the user's hard drive, as specified by *DST.* Results will be logged to the screen and/or the log file, depending upon the settings of the *SILENT* and *LOG* commands. Files that exist only in the PQI and not on the HD will be identified as *missing*. No entry will be made for files that exist only on the hard drive. For files that exist in both places, those that are byte-for-byte the same will be listed as *identical*, while the others will be logged as *changed*.
COPY		Copies all files in the PQI file as specified by *SRC* to the user's hard drive, to the directory specified by *DST.* The display and/or log, as defined by the *silent* and *log* commands will list each file that has been extracted from the PQI, indicating that it has been *restored*.
CRC	<yes/no>	*DIR* option. Computes a CRC for each file that is listed. Computing the CRC requires that the file be read from the PQI, so this option makes Listing much slower.
CREATE	<yes/no>	*DIR* option. Shows the creation date/time for each file listed. This data may not always be available on FAT partitions.
DIR		Lists all of the files and directories contained in the PQI file as specified by the argument to *SRC*. Include the size of each file. If the *SUB* option is set to *true*, then also list the files in each sub-directory. The commands *CRC, CREATE, ACCESS, MODIFY* and *SAT* can be used to change the information shown during a listing.

Command	Parameter	Description

DST	<path>	Destination path (on hard drive). Specifies the drive and directory where the files should be restored or compared. If the directory does not exist it will be created. This can be either a relative or absolute path. This command must specify a directory and not a file, even if only a single file is to be restored. To specify a hidden partition, you should use the PQAccess number for that partition in place of the drive letter. The *INFO* command will display the PQAccess numbers for each partition. For example if the first partition on the first drive is an NTFS partition, and you are running from DOS (booted off a floppy), the DOS will not know about the partition or assign it a drive letter, but PQIEXTD.EXE can still restore files to it by specifying:
		`-DST=1:\`
		You can also identify a path using a volume label. For example, you could point to a "WIN98" volume using the following command:
		`-DST=:WIN98\Windows`

ECHO	<text>	Echoes the specified text to the screen and/or log file. If *silent* mode is not enabled, then the text will be displayed on the screen. If logging is enabled it will also be written to the log file. This would most often be used to display some sort of status information to the end-user.
ERR	<file>	Records exit status to file. Upon completion of the application, the error file will only exist if the application encountered an error. If a file with this name existed before the application ran, it will be deleted. The error file is a text file. The first line of the error file contains the error code (number). The next line in the file contains a description of the error. If multiple errors occur, only the first error will be recorded in this file. All errors will be logged to the log file, if logging is enabled.
EXIT		Exits immediately. This command is usually not needed. It exists so that if all commands are specified within the RTC file, PQIExtract can be commanded to exit immediately after completing the RTC commands, without processing any commands from the command line or from a script file.

Command	Parameter	Description
IDE	<yes/no>	*(DOS only)* When accessing an IDE hard disk using PQ direct methods as opposed to OS calls, the disk is normally accessed using BIOS interrupt 13. This switch uses UDMA accessess instead, which is generally much faster. However, UDMA access does not always work, especially on older systems. This command is only available in the DOS version. It is disabled by default.
INFO		Shows information about each partition and free space on all system hard drives. The first two columns display the disk number and partition number as used in Drive Image scripting. The third column specifies the drive letter assigned to the partition if any by the current OS. The fourth column specifies an alternate method of referring to each partition with a “drive letter” that is really a number, which is called the PQAccess number. This PQAccess number (including the colon), can be specified to PQIExtract as part of the path argument to the *DST* command, in place of the drive letter. This is how to specify to PQIExtract to restore files to partitions not visible to the OS. The remaining columns display the partition size, the amount of used space within the partition, its type and its label. This is the same information displayed when using the /INFO switch with Drive Image.
LEF	<yes/no>	When extracting files, this option controls whether or not the name of each file is logged after it has been extracted, updated or compared. If enabled, each file is logged to the screen and/or log file, depending upon the settings of the *SILENT* and *LOG* commands, as it is processed. If disabled, then filenames are not logged, unless an error is encountered while processing that file. When using the compare command, the names of changed files will always be logged, regardless of the setting of this command. This command is mainly useful in reducing the size of the log file when a large number of files are going to be extracted. Enabled by default.
LOG	<file>	Logs information to file. Log information about the status of all commands to a file. All error information will be recorded in the log file. If a file with this name existed before running, it will be overwritten.
MODIFY	<yes/no>	*DIR* option. Shows the last modification date and time for each file listed. Enabled by default.

Command	Parameter	Description
NOX		Clears all of the exclude lists created by *XDR, XFL, XWC* or *XAT*, so that all files are again listed and restored.
PQI	<file>	Uses specified PQI file. It is possible to use multiple PQI files, extracting files first from one PQI file and then additional files from a second PQI, but this case is not common.
PROGRESS	<bytes>	PQIExtract cannot normally show percent done when extracting files, because it does not know how much data it will be pulling from the PQI file. However, PQIExtract reports the number of bytes that it has extracted when it completes. Therefore, you can use this number as an argument to *PROGRESS* the next time that you run PQIExtract with the same options and PQIExtract will display its progress (percentage completed) as it extracts files.
PWD	<password>	Provides PQI password. If a partition within the PQI is encrypted, you can use this command to provide the password. You must specify this command before you specify the name of the PQI file (using the *PQI* command). The password command may be called multiple times. The last five passwords will be remembered. PQIExtract will be able to access an encrypted PQI file or partition if any of these five passwords is correct.
RESTORE		Restores changed files. This command behaves like *CMP*, except that when it finds a file that is missing or changed, it copies it to the hard drive. Therefore the log will only show a status of *identical* or *restored*. This command is generally slower than *COPY.*

Command	Parameter	Description
SAT	<yes/no>	*DIR* option. Shows the file attributes. Enabled by default. For FAT, FAT32 and NTFS partitions, the attributes are: R = Read-Only H = Hidden S = System A = Archived D = Directory C = Compressed (NTFS) A file may have multiple attributes. For Ext2 and Ext3 partitions, the attributes are always displayed as five characters. The possible first character values are: D = Directory S = Socket L = Symbolic Link F = FIFO B = Block Device C = Character Device ‘ ‘ = Normal File The next three characters are the digits 0-7 representing the UNIX access rights to the file. The first character is the owner’s rights, the second character is the group’s rights, and the third character is other’s rights. Bit zero of each digit is execute permission (x), bit one is write permission (w) and bit two is read permission (r). Standard Linux notation uses ‘-‘ when a permission is not granted, so: 0 = `---` No Rights 1 = `--x` Execute Only 2 = `-w-` Write Only 3 = `-wx` Write/Execute 4 = `r--` Read only 5 = `r-x` Read/Execute 6 = `rw-` Read/Write 7 = `rwx` Read/Write/Execute The last character in the Ext2 or Ext3 attributes field will be an ‘R’ if the file is owned by root; otherwise it will be blank.

Command	Parameter	Description
SILENT	<yes/no>	Silent mode. If *SILENT* mode is enabled, most text messages are not displayed on the screen. Progress and error messages will continue to be displayed. By default, silent mode is disabled.
SRC	<path/file>	Source path or file in the PQI file. Specifies a file or directory contained within the PQI file, which should be extracted, compared or listed (depending upon which of these commands is later specified). If a file is specified, then only that single file will be extracted. If a directory is specified, then all files in that directory will be extracted. If the *SUB* mode is enabled, then all files in all subdirectories of the specified directory will also be extracted. If a PQI file contains images of multiple partitions, then by default it will look for the path in the first partition in the PQI (partition0). If you want to specify a different partition, then proceed that path with `\partitionX`, where X is the partition number. For example, `\partition1\windows` will extract all files in the windows directory of the second partition. The file portion of <path/file> can include the wildcard characters "*" and "?". An asterisk matches a group of characters. A question mark matches a single character. All files in the directory specified by the path portion of <path/file> that match the wildcard mask will be extracted. If *SUB* mode is enabled, the mask will also be applied to all files in all subdirectories below the specified starting directory and all matching files will be extracted. For example, the command "`SRC = c:\windows\options\cabs\*.cab`" will only extract CAB files located in the c:\windows\options\cabs directory. Note that to get all files use "`*`" not "`*.*`". The second method will only access files that have a period in their name.
SUB	<yes/no>	Recursively processes subdirectories. If *SRC* is a directory, you can use this option to extract all subdirectories of the specified directory. By default subdirectories are not processed.

Command	Parameter	Description
USE-OS	<yes/no>	If the partition specified using the *DST* command is visible to the operating system (generally this means that it has been assigned a drive letter), this commands instructs the application to use OS calls to perform the commands specified in the PQI file. When accessing a hidden partition, the setting of this command is ignored, because PQIExtract cannot use OS calls to access the partition. Unless the target partition has recently been modified without the OS's knowledge, perhaps by a program such as Drive Image or Norton PartitionMagic, then you should leave this command enabled. **Use great care if you wish to disable this command.** If the OS has not flushed all of its caches before PQIExtract runs, or if the OS attempts to access the target partition after PQIExtract has modified it "behind its (the OS's) back" without first rebooting, then the partition may become corrupted. By default this command is enabled.
VER		Shows version information. If logging is enabled, this information will be recorded in the log. This information will display to the screen only if *SILENT* mode is not enabled.

XAT	<attribs>	Excludes files that have or don't have the specified attributes. Directories attributes are not checked. Possible attributes:
		R = Read-Only
		H = Hidden
		S = System
		A = Archived
		C = Compressed
		N = Normal
		N is a special case. When used, it must be the only attribute specified and causes all files to pass the attribute test (default). N is useful for clearing a previous attribute, without using the *NOX* command.

Command	Parameter	Description
XAT *(continued)*		Specifying a minus before an attribute means exclude any file that does not have that attribute. The minus affects all following specified attributes, until a plus sign is seen. Specifying a plus or not specifying any sign before an attribute means to exclude any file that has that attribute. For example, `/XAT HS-R` excludes from being listed/extracted, any file that is a hidden file or a system file or not read-only.
XDR	<dirs>	Exclude the specified directory or directories from being listed or extracted. The calls are cumulative, so that if *XDR* is called multiple times each directory is added to the list of excluded directories. It is also possible to specify multiple directories at the same time, by separating each directory with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the directory name. The list of excluded directories can only be cleared by a call to *NOX*. The names specified in this command must be either the full path to the directory or the relative path to the directory from the directory where the extract started. For example, if extracting with `/SRC= \Partition0\Windows\` and you want to exclude \Partition0\Windows\system32\ drivers, you must specify `/XDR \Partition0\Windows\system32\ drivers` or `/XDR system32\drivers.` The specified directory names are compared to the directory names in the PQI file, not to the name on the hard drive.

Command	Parameter	Description
XFL	<files>	Excludes the specified file or files from being listed or extracted. The calls are cumulative so that if *XFL* is called multiple times, each file is added to the list of excluded files. It is also possible to specify multiple files at the same time, by separating each filename with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the filename. The filename must include the full path of the file, and cannot include any wildcard characters. The list of excluded files can only be cleared by a call to *NOX*. The names specified in this command must be either the full path to the file or the relative path to the file from the directory where the extract started. For example, if extracting with /SRC=`\Partition0\Windows\` and you want to exclude Windows\system32\desktop.ini, you must specify `/XDR\Partition0\Windows\system32\desktop.ini` or `system32\desktop.ini`. The specified filenames are compared to the path/filenames in the PQI file, not to the path on the hard drive.
XWC	<masks>	Excludes any files whose filename matches any of the specified wildcard mask from being listed or extracted. The calls are cumulative so that if *XWC* is called multiple times, each mask is added to the list of masks. It is also possible to specify multiple masks at the same time, by separating each wildcard mask with a semi-colon (;). Spaces are not allowed on either side of the semi-colon, unless they are part of the mask. The masks cannot include directory information, as they are only applied to the filename portion of each file, and not to the full path name. The list of excluded masks can only be cleared by a call to *NOX*.

Command Aliases

You can substitute commands to use any of the following aliases.

Command	Alias
?	h help
ABT	allow-abort allow-ctrl-c
ACCESS	acc access-date
AFX	auto-fix
CMD	script script-file
CMP	comp compare
COPY	cp cpy
CRC	show-crc compute-crc
CREATE	cre create-date
DIR	lst list
DST	dest-path hd-path
ECHO	prn print
ERR	error error-file
EXIT	die terminate
IDE	udma enable-udma
INFO	inf partition-info
LEF	log-files log-each-file
LOG	log-file
MODIFY	mod modify-date
NOX	xno no-x x-none no-exclude exclude-none
PQI	img image pqi-file image-file
PROGRESS	prg show-progress
PWD	password
RESTORE	upd update
SAT	show-attr show-attrib
SILENT	sil quiet
SRC	src-path pqi-path
SUB	rec recursive
USE-OS	uos use-os-calls
VER	version
XAT	x-attr x-attrib exclude-attr exclude-attrib
XDR	x-dir exclude-dir
XFL	x-file exclude-file
XWC	x-wc x-wildcard exclude-wc exclude-wildcard

Script File Example

In the following example, **E:** is assumed to be a CD drive containing a CD with a PQI file named **FACTORY.PQI**. PQIExtract is used to extract all of the changed files in the **\Windows\options\cabs** directory, to a directory of the same name on the `C:` drive.

A typical script file called **GetCAB.PQE** located on **E:** looks like the following:

```
// GetCAB.pqe
// Restore all the CAB files to
// C:\Windows\options\cabs
// so Windows can be reinstalled

// Disable most messages to the screen
Silent

// Record results to a file in case it is needed by
// tech support
Log = c:\extract.log

// Add PQIExtract version information to the log file.
Version

// Specify the PQI file
PQI-File = e:\Factory.pqi

// Specify the location within the PQI of files
// to be extracted.
SrcPath = \windows\options\cabs

// Specify the location to place the extracted files.
DestPath = c:\windows\options\cabs

// Extract files that have changed or been deleted.
Restore

// End of file
```

To run the application, use the following command:

```
PQIEXTD /cmd=e:\getcab.pqe
```

When a file is restored, the original file attributes and file times are also restored. This includes file attributes and creation, last modification and last access times/dates for each file that it restores. Because of inconsistencies in DOS, PQIEXTD does not always handle times perfectly, especially with regard to time zones and daylight savings time. As a result, time values set by PQIEXTD may vary slightly from the original times (when not correct, they are generally off by exactly one hour). PQIEXTD may exhibit the same time inconsistencies when listing file times. The following attributes will be restored: Hidden, System, Read-only and Archived.

Note that NTFS supports additional attributes that are not restored, and no attributes are restored for Ext2 or Ext3 files.

RTC Files

A runtime configuration file (RTC) is supplied for each version of PQIExtract. The RTC file has the same name as the corresponding executable, except the extension is .RTC instead of .EXE. Any PQIExtract command or group of commands can be added to the RTC file, so that these commands are executed every time the application runs. A common command to embed in the RTC file is the PQI file password. This hides the password from customers and protects the PQI file from restoration, except by those with access to the proper RTC file.

Any PQIExtract command can also be disabled in the RTC file, so that the command is not available to customers, preventing them from using PQIExtract in ways not intended. It is also possible to hide commands so that they don't display when the *help* command is used, however, they can still be called.

PQIExtract also has the ability to "lock" itself to run only on computers with a specific BIOS or DMI string. BIOS information is specified in the RTC file. When locking is enabled, the system BIOS or DMI is checked before PQIExtract is allowed to execute. If the lock test fails, a customizable error message is displayed indicating the failure. If BIOS-locking is enabled and the PQI password is specified in the RTC file, the image is locked so it can only be restored on a machine with the appropriate BIOS.

CHAPTER 5

PQA

Overview

PQA enables you to update, add, and delete selected files or directories on a visible or hidden partition on a hard disk. This system can quickly replace device driver files, provide patch files, or make other file change operations to the partition without having to build a new PQI image file. Since a PQA file contains only file modification commands and file data, it can be used with any applicable partition.

PQA consists of two programs: PqaBuild and PqaDeploy. The PqaBuild program creates image addendum files by packaging and compressing all the specified file modification commands and file data into one PQA file. The PQADeploy program is then used to apply the PQA file modification commands and file data to the specified visible or hidden partition. You can run either program in one of two modes: command line mode or script mode.

PQA is analogous to the ZIP system, where a single ZIP file is created from a collection of files. This one ZIP file can then be used to restore all the files to the target destination on many different systems. However, in addition to just specifying files, a PQA file can also contain commands to delete and rename files and create and delete directories.

The real power of PQA lies in its ability to modify both visible and hidden partitions. By default, when the target partition is visible to the running operating system, PqaDeploy uses the OS I/O system to make the file modifications. When the target partition is hidden to the running operating system, PqaDeploy uses the Symantec file system engine to make the file modifications.

The access method used to modify the target partition can be specified explicitly in the PqaDeploy script file or on the command line. Care must be taken when using the Symantec file system engine to modify a visible partition so that the OS does not unknowingly overwrite the PQA directed modifications.

There are three versions of the PQA file builder and deploy programs for multiple platforms:

- PqaBuild9x/ PqaDeploy9x – Windows 9x or Windows Me
- PqaBuildNT/ PqaDeployNT – Windows NT, Windows 2000, Windows PE, and Windows XP

- PqaBldD/ PqaDplyD – DOS version

All Builder versions run in a DOS box on Windows 9x, Me, PE, NT or 2000. However, PqaBldD does not support long filenames, since it uses DOS calls to access files.

Both Windows deploy versions run in a Windows DOS box for the appropriate OS and support long filenames. PqaDplyD only runs on DOS, but it will restore long filenames if they are present in the PQA file. Unless otherwise noted, all references and examples of one version also apply to the other versions.

If an error occurs while running, PqaBldD and PqaDplyD will exit with an exit code of one (1). If an error occurs while running, PqaBuild9x, PqaBuildNt, PqaDeploy9x and PqaDeployNt will exit with an exit code that is the same as the error number. If no error occurs, all will exit with a value of zero.

Running PqaDeploy and PqaBuild

To run PqaDeploy or PqaBuild, specify the program name followed by optional arguments. For example:

```
PQABLDD {argument, [argument]…}         (DOS command prompt)
PQABUILD9x {argument, [argument]…}      (Windows 9x DOS box)
PQABUILDNT {argument, [argument]…}      (Windows NT DOS box)

PQADPLYD {argument, [argument]…}        (DOS command prompt)
PQADEPLOY9x {argument, [argument]…}     (Windows 9x DOS box)
PQADEPLOYNT {argument, [argument]…}     (Windows NT DOS box)
```

IMPORTANT! PqaBuild aborts immediately when <Ctrl-C> is pressed. The PQA file being created is usually corrupt as a result. PqaDeploy does not abort until it is done processing the current file, and a command line option allows <Ctrl-C> to be ignored completely.

Both programs may be run in either of two modes, command line mode or script mode. In command line mode, you specify all commands on a single command line. In script mode, you specify commands in a script file and the command line specifies just the script filename. Using script mode overcomes the limitations in the length of the command line and allows you to perform more complex operations.

Command Syntax

When used on the command line, commands are preceded by either a forward slash (/) or a minus sign (-). Commands may be shortened to use any of the abbreviations or synonyms listed in the table below. Commands are not case-sensitive, and the embedded minus signs (-) can be replaced with underscores (_) or omitted completely. For example, the following are all equivalent commands:

```
log-file
LogFile
LOG_FILE
```

```
log
```

On the command line, program commands may be followed by an equal sign followed by an argument in order to pass a value to the command. The argument may also be proceeded by a space, instead of an equal sign, but in this case, the argument must not start with a "/" or "-" or it will be interpreted as the next command, instead of as an argument to the previous command. For example,

```
PQABuild /Silent=yes
PQABuild /Silent = yes
PQABuild /Silent yes
```

are all valid. Spaces between arguments will be replaced with a single space, so if you are attempting to pass a filename or other argument that contains multiple spaces, then you must place the argument in quotes so that the spaces are not eliminated.

To run these programs in script mode, specify a single command to load a script file, and then specify the remaining commands from within the script file. A script file can have any extension, but ".PQB" is the recommended extension for PqaBuild script files, while ".PQD" is the recommended extension for PqaDeploy script files.

Script files are text files that have one command on each line. Blank lines, and lines that start with a ";" or a "#" or a "//" are ignored and can be used for comments within the script file. In script files, commands are not preceded by a forward slash or a minus sign as they are on the command line. Spaces are allowed before and after the command and on either side of the equal sign. The equal sign is optional. For example, in a script file the following commands are equivalent to one another:

```
Silent=yes
  Silent   =   yes
Silent yes
```

Command Order

Command order is important, both on the command line and in a script file.

```
PqaDeploy /pqa=file.pqa /drive=1 /partition=1 /deploy
```

applies the PQA commands in FILE.PQA to the specified partition, but

```
PqaDeploy /deploy /pqa=file.pqa /drive=1 /partition=1
```

will not, because the PQA file, target drive number and partition number have not yet been specified when the deploy command is executed.

IMPORTANT! If you want results to be recorded in a log file, you must specify the *LOG* command before specifying the other commands.

Accessing Hidden Partitions

PqaDeploy assigns a partition number to every partition on all hard drives on the system. Free space on the drives may also be assigned a number. These numbers can be used in place of drive letters to access partitions. Use the /info command to get a listing of all partitions along with their numbers.

The following is an example of the information that would display if you ran `PqaDeploy /info`.

```
Partition Information
Disk Part # Letter PqAccess Size(MB) Used(MB) Type        Label
==== ====== ====== ======== ======== ======== =========== ===========
  1     1     C:        1:     2502     1209 FAT32       WIN98
  1           *:        2:     5554     5554 Extended
  1     2     *:        3:     5554     3531 NTFS        Win2K
  2     1     *:        4:        8        2 Free
  2           *:        5:    17124    16865 Extended
  2     2     D:        6:     2039     1924 FAT32       SOURCE
  2     3     E:        7:     4040     1755 FAT32       PQI DATA
  2     4     *:        8:     6001     4651 NTFS        NTFS-2
  2     5     F:        9:     1412       55 FAT         FATTEST
  2     6     G:       10:     3067     2566 FAT32       FAT32BACKUP
  2     7     *:       11:      204        3 NTFS        NTFS-Hidden
  2     8     *:       12:      102       28 FAT32       FAT32-HDN
  2     9     *:       13:      259        0 Free
  2    10     *:       14:     2432        0 Free
```

This example shows two hard drives with a FAT32 and an NTFS partition on the first drive and a collection of partitions on the second hard drive. Running under DOS, only the FAT and FAT32 partitions are assigned drive letters. The third column lists the assigned drive letter if there is one. The fourth column lists the drive number that can be used by PQAccess in place of the drive letter to access FAT, FAT32, and NTFS partitions. The drive numbers can also be used by PqaDeploy to access the corresponding partition. Refer to the description of the /info command on page 66 for more information.

To deploy a PQA to a hidden NTFS partition, you could use the following command:

```
PqaDeploy -pqa=data.pqa -ltr=3: -deploy
```

The command above is equivalent to the command:

```
PqaDeploy -pqa=data.pqa -drive=1 -part=2 -deploy
```

Command Parameters

Command parameters are specified by placing a space or an equal sign (=) after the command, followed by a parameter value. The following table describes the available command parameters.

Parameter	Description
<file>	This indicates a filename, optionally including a drive letter and an absolute or relative path to the file. If a path is specified, then all directories in the path must already exist. The *CMD* command requires that the file already exist, and will cause an error if the specified file cannot be found. The same is true of the *PQA* command for PqaDeploy, but with PqaBuild, if the file specified by PQA already exists, it will be overwritten. The commands *LOG* and *ERR* will create new files with the specified name. If the file already exists, the existing file will be deleted and replaced by the new one.
<pfile>	Similar to <file> but cannot include a drive letter, and must be an absolute path. It is used to refer to a file or directory on the target machine of PqaDeploy.
<yes/no>	For commands that take a <yes/no> parameter, the parameter is optional. If not specified, then *yes* is assumed. Several synonyms are also available. The strings “yes” “y” “on” “true” “t” and “enable” are all equivalent and enable the specified command mode. The strings “no” “n” “off” “false” “f” and “disable” all disable the specified command mode. These strings are not case-sensitive.
<password>	A valid password string. This string is unencrypted, so your password may be revealed if the customer can see this command.

Parameter **Description**

<src>[,<dest>]

<src> specifies a file or directory on the source machine (machine used when building the PQA file). It must contain drive letter specifications (or PQAccess partition numbers) as the first two characters and may use wildcard characters ‘*’ and ‘?’ as part of the filename, but not as part of the path. Wildcard characters will only match files, not directories.

<dest> specifies a file or directory. The directory can be on the target machine or a network. <dest> should never include a drive letter. If <src> specifies a directory, or uses wildcards, then <dest> must specify a directory. If <src> specifies a single file, then <dest> can specify a file or a directory. For either <src> or <dest> to specify a directory, the last character of the string must be a backslash (\) or a forward slash (/).

<dest> is optional. If <dest> is not specified, then <src> is also used as the <dest> parameter (except that the drive letter portion is dropped, and when wildcards are used, the filename portion is dropped so that only the directory portion is used).

The comma between <src> and <dest> is optional and may be replaced by a space. If <src> contains embedded spaces or commands, it should be placed between either single (’) or double (") quotes. <dest> may also be placed in quotes, but they are never required. For example,

```
  C:\program files\prog.exe, \program files\newname.exe
```

would incorrectly assign

```
  <src>="C:\program"
```

and

```
  <dest>="files\ prog.exe, \program files\newname.exe"
```

Instead use:

```
"C:\program files\prog.exe", "\program files\newname.exe"
```

<pdir>

This specifies a directory on the target machine. It must be an absolute path beginning with a forward or backward slash, “/” or “\”.

<letter:>

An operating system visible partition specifier, such as “A”, “A:” or if the OS supports it, “\\machine\dir” where *machine* is a network machine and *dir* is a mount point on the machine. Alternately, you can use a PQAccess partition number or a volume label.

<number>

A number.

Commands Common to Both PqaBuild and PqaDeploy

The following table lists the commands common to both PqaBuild and PqaDeploy. For a list of command abbreviations, see "Command Aliases" on page 68. For information about command parameters, see "Command Parameters" on page 59.

Command	Parameter	Description
?		Display a brief summary of all enabled commands. All of the commands in this list that have not been disabled in the RTC file will be displayed. This command can take a number as an optional parameter <n>. If specified and the help is being displayed to a window (that is, *SILENT* has not been specified), then the output will pause after each <n> lines of output are displayed. Use a value of zero for <n> to disable pausing. Default value for <n> is 22.
AFX	<yes/no>	Automatically fix partition table errors, such as the CHS values not matching the LBA values. Disabled by default.
CMD	<file>	Read commands from a script file. The script file must exist. All commands in the script file will be executed before any more commands from the command line are processed. See "Command Syntax" on page 56 for information about the format of commands in a script file.
ECHO	<text>	Echo the specified text to the screen and/or log file. If *SILENT* mode is not enabled, the text will be displayed on the screen. If logging is enabled it will also be written to the log file. This would most often be used to display some sort of status information to the end-user.
ERR	<file>	Record exit status to file. Upon completion of the application, the error file will only exist if the application encountered an error. The error file is a text file. The first line of the error file specifies the error code, which is also the exit code of the application. The next line in the file contains a description of the error. If multiple errors occur, only the first error will be recorded in this file. If no error occurs, this file will not exist after the program exits. If a file with this name existed before running, it will be deleted.

Command	Parameter	Description
EXIT		Exit immediately. This command is not usually needed. It exists so that if all commands are specified within the RTC file, PqaBuild or PqaDeploy can be commanded to exit immediately after completing the RTC commands without processing any commands from the command line or script file.
LOG	<file>	Log information to file. Log information about the status of all commands to a file. All error information will be recorded in the log file. If a file with this name existed before running, it will be deleted. The *LOG* command should generally be the first command on the command line or in the script.
PQA	<file>	Use specified PQA file. It is possible to use multiple PQA files, extracting files first from one PQA file and then additional files from a second PQA; however, this case is not common.
PWD	<password>	Provide PQA password. You can use this command to provide the password. You must specify this command before you specify the name of the PQA file (using the *PQA* command). The password command may be called multiple times. The last five passwords will be remembered. PqaDeploy can access an encrypted PQA file if any of these five passwords is correct. PqaBuild will always use the last specified password.
SILENT	<yes/no>	Silent mode. If *silent* mode is enabled, most text messages are not displayed on the screen. Progress and error messages will continue to be displayed. By default, silent mode is disabled.
VER		Show version information. If logging is enabled, this information will be recorded in the log file. This information will display to the screen only if *SILENT* mode is not enabled.

Commands Specific to PqaBuild

The following table lists the commands specific to PqaBuild. For an explanation of command parameters, see "Command Parameters" on page 59.

Command	Parameter	Description
CHECK	<pfile>	When restoring, the specified file or directory must already exist on the target system or an error will be raised.
CMP	<comp-type>	Specify the compression type used to store the PQA file. If used, this must be specified before the *PQA* command is specified. Valid values for <comp-type> are *none, low, high, zlow, zhigh, hlow* and *hhigh*. These strings are not case-sensitive. *Low* and *high* use the default compression type, which is currently *hlow* and *hhigh*. *Zlow* and *zhigh* are based on a different compression algorithm that is slower at both compression and decompression but generally produces smaller files.
COPY	<src>[,<dest>]	Read <src> file from local drive and include in PQA with the path and name specified by <dest> if present. If <dest> only specifies a directory, then it will use the filename from <src>. If <dest> only specifies a filename, then the file will be placed in the same directory as <src>. If <dest> is not specified at all, then the destination path will be the same as the source path. The <src> specification can include wildcard characters '*' and '?', which will be used to match against all files in the specified directory on the local disk, adding each matching file to the PQA. Wildcards will not match directories. If <src> is a directory, then all files in that directory will be added to the PQA. If <src> is a directory, or uses wildcards, and the *SUB* option has been enabled, then all subdirectories of the specified directory will also be processed. If <src> is a directory, or uses wildcards, and <dest> is specified, then <dest> must be a directory. The <dest> directory must already exist on the target machine. <dest> should never include a drive letter, as the target partition is specified at restore time to PqaDeploy.

Command	Parameter	Description
CRC	<src>[,<dest>]	Compute the CRC of <src> located on the local system, and record the CRC and in the PQA. When restoring the PQA, compute the <dest> file's CRC and compare to the stored CRC. If the <dest> file is not present, or if the CRCs do not match, then generate an error. This effectively tests for a particular version of a file. It can be useful to verify that the PQA file is being applied to the appropriate partition.
DEL	<pfile>	Delete the specified file. Full path name should be specified to the file. The filename portion of the <pfile> specification can include the wildcard characters '*' and '?'. The wildcard characters will not be expanded until PqaDeploy is used, so they will match files on the target system. If <pfile> specifies a directory, then all files and subdirectories of the directory will be deleted. Wildcard matches cannot be used to match directory names. PqaDeploy will not generate an error if the file does not exist when it goes to delete it, but it will generate an error if it is unable to delete an existing file. Use the *CRC* or the *CKECK* command if you want to verify that a file exists before deleting it.
MFS	<size>	Specify the maximum size (in bytes) that the PQA file can be before spanning (that is, splitting data among multiple files). If used, this command must be specified before the *PQA* command is specified. The minimum value for <size> must be at least 1,048,576 (1 MB). The size can be followed by the letters "K" for kilobytes or "M" for megabytes. The following are all equivalent: "/MFS=1048756," "/MFS=1024K," and "/MFS=1M."

Command	Parameter	Description
MKDIR	<pdir>	Store a command in the PQA to create a directory. All directories must exist on the destination system before a file can be restored to that directory. This command should be used before a *COPY* or *NEW* command that writes a file to a potentially none-existent directory. When a directory and all of its subdirectories have been copied or added to a PQA, the subdirectories will be automatically created if they do not exist; thus, this command is not needed in that case. However, it will still be necessary to create the starting directory if it does not exist on the target. When a PQA contains the *MKDIR* command, no error will be generated if the directory already exists. This will be noted in the log file if any.
NEW	<src>[,<dest>]	Same as *COPY*, except that when restoring the file, the file must not already exist on the target machine. It is an error if the file already exists, which will stop the deploy. **Only use this if you know the file will not exist and want an error if it does**; otherwise, use the *COPY* command.
REN	<pfile1>,<pfile2>	Rename the <pfile1> so that it has the name <pfile2> name. If <pfile2> does not contain any backslashes, "\", then the file will be renamed. Otherwise the file will be moved to the location specified by the path portion of <pfile2>. The new path may be an absolute or a relative path.
SUB	<yes/no>	Recursively process sub-directories when performing the *COPY*, *NEW* or *UPDATE* commands and when the <src> is a directory or uses wildcards. By default, subdirectories are not processed.
UPDATE	<src>[,<dest>]	Same as *COPY*, except that when restoring the file, the file must already exist on the target machine. It is an error if the file does not exist, which will stop the deploy. **Only use this if you know the file will exist and want an error if it does not**; otherwise, use the *COPY* command.

Commands Specific to PqaDeploy

For an explanation of command parameters, see "Command Parameters" on page 59.

Command	Parameter	Description
DEPLOY		Apply the changes specified in the PQA. The target partition must already have been specified using either the *LETTER* command or both the *DRIVE* and *PARTITION* commands. If all are specified, the *LETTER* command setting will be used.
DRIVE	<number>	Select a drive. Drive 1 is the first disk in the system. Number is the same as for Drive Image scripting. All IDE disks occur first, followed by SCSI disks, if any.
IDE	<yes/no>	*(DOS only)* When accessing an IDE hard disk using PQ direct methods, as opposed to OS calls, the disk is normally accessed using BIOS interrupt 13. This switch uses UDMA accesses instead, which is generally much faster. However, UDMA access does not always work, especially on older systems. This command is only available in the DOS version. It is disabled by default.
INFO		List information about all partitions on the local system. The first two columns of output include the drive number and partition number that can be used by the *select-drive* and *select-partition* commands for selecting a particular partition. The next column displays the drive letter associated with the partition, if the partition is visible to the currently running OS. "*:" will be displayed if the drive letter is not visible to the OS. The fourth column displays the partition numbering convention used by the Symantec PQAccess application. The *LETTER* command can specify either the drive letter from column three or the PQAccess naming from column four. The remaining columns display the partition size, used space, type, and volume label.
LETTER	<letter:>	Select both the drive and partition for deploying the PQA. <letter> should specify either the PQAccess partition number followed by a colon (see INFO above for details) or the drive letter (optionally followed by a colon) or a mount point. Example of <letter> are "C", "C:", "3:", or "\\machine\dir." You can also specify a subdirectory on the specified drive, such as C:\TEST\.

Command	Parameter	Description
LIST	{short}	List all commands contained in the PQA. Show files that will be deleted, files that will be added, and so forth. If you include the optional {short} parameter, short filenames will be listed along with long filenames.
PARTITION	<number>	Select a partition on the selected drive. Partition 1 is the first partition on the drive. Partitions are numbered sequentially on the drive. Extended partitions are not numbered. Numbering is the same as for Drive Image scripting.
REBOOT	<yes/no>	Reboot the computer when the program exits. Disabled by default. This command does not work on Windows 9x systems.
USE-OS	<yes/no>	If the specified partition is visible to the OS (generally this means that it has been assigned a drive letter), use OS calls to perform the commands specified in the PQA file. Unless the target partition has recently been modified without the OS's knowledge, perhaps by a program such as Drive Image or Norton PartitionMagic, then you should leave this command enabled. Use great care if you wish to disable this command. If the OS has not flushed all of its caches before PqaDeploy runs, or if the OS attempts to access the target partition after PqaDeploy has modified it "behind its (the OS's) back" without first rebooting, then the partition may become corrupted. By default this command is enabled.

Command Aliases

You can substitute commands to use any of the following aliases.

Command	Aliases
?	h help
AFX	auto-fix
CHECK	chk exists
CMD	script script-file
CMP	compress compression
COPY	cp cpy
CRC	show-crc compute-crc
DEL	delete
DEPLOY	dep restore
DRIVE	drv select-drive
ECHO	prn print
ERR	err error error-file
EXIT	die terminate
IDE	udma enable-udma
INFO	inf partition-info
LETTER	ltr drive-letter
LIST	lst list-commands
LOG	log-file
MFS	span-size max-file-size
MKDIR	md make-dir
NEW	add
PARTITION	prt select-partition
PQA	pqa-file
PWD	password
REBOOT	rbt
REN	move rename
SILENT	sil quiet
SUB	rec recursive
USE-OS	uos use-os-calls
UPDATE	upd replace
VER	version

Examples

Both Pqabuild and PqaDeploy may be run in either of two modes: command line mode or script mode. Below are two examples. The first example uses both programs in command line mode and the second example uses both programs in script mode. The two modes can also be mixed depending on your needs.

Command Line Mode Example

The first example is using PqaBuild in command line mode to build a PQA file to replace one driver file. The PqaBuild program can be called from the command line as follows:

```
PqaBuild /PQA=NET2K.PQA /Delete \windows\system\net9x.sys /Copy
c:\temp\net2k.sys \windows\system\net2k.sys
```

This specifies that the PqaBuild program create a PQA filenamed NET2K.PQA. Then it places the “DELETE \windows\system\net9x.sys” command into the PQA file. Next, a copy of the file data found at c:\temp\net2k.sys is placed in the PQA with a command to restore this file to \windows\system\net2k.sys during the deploy step.

The deploy disk and partition numbers are not specified in the PQA file, but are specified in the deploy step. This allows one PQA file to be used on different disk and partition configurations.

When the PQA file has been built, the PqaDeploy program can then be called from the command line to run the commands contained in the PQA file as follows:

```
PqaDeploy /PQA=NET2K.PQA /DRIVE=1 /PARTITION=1 /DEPLOY
```

This will execute the commands and apply the files contained within the NET2K.PQA file to the first partition on the first disk. This simple example demonstrates the command line mode and is limited to the commands that can be specified on one command line.

To accomplish more complex tasks, use the PqaBuild script mode. This allows you to specify an unlimited number of commands and files to be included in the PQA file using a PqaBuild script file. The deploy options can also be specified in a PqaDeploy script file.

Script Mode Example

Below is an example of building a PQA file to replace two driver files. The PqaBuild and PqaDeploy script files, and the command lines used to run the programs are listed.

For PqaBuild, a minimal script file contains two commands, the name of the PQA file to create, and a file to add or delete.

Sample PQABuild Script

The following is a sample PqaBuild script file, called NET2K.PQB.

```
// NET2K.PQB
// Replace the network driver with an updated version
// Disable most messages to the screen
Silent
// Record Results to a file in case it is needed by
// tech support
Log = c:\build.log

// Specify the PQA filename
PQA = NET2k.PQA

// Replace files that have changed or been deleted
Del \windows\system\net9x.sys
Del \windows\system\inf\net9x.inf
Copy c:\temp\net2k.sys \windows\system\net2k.sys
Copy c:\temp\net2k.inf \windows\system\inf\net2k.inf

// End of file
```

To run the builder in script mode, use the following command line:

```
PqaBuild /cmd=net2k.pqb
```

This creates a file called NET2K.PQA in the current directory. When this PQA file is deployed, it will delete the file net9x.sys in the \windows\system directory and the net9x.inf file in the \windows\system\inf directory. It will then write the new versions of these files captured within the PQA file into the specified locations with new versions of the files that were read from the c:\temp directory when the PQA file was built.

PqaDeploy can then be used to deploy the PQA file to a hard drive partition. To run PqaDeploy in command line mode, use the following command. (The line wrapping shown here is incidental to the manual only. Commands must be contained on a single line.)

```
PqaDeploy /log=f:\deploy.log /pqa=f:NET2K.PQA /silent /drive=1 /partition=1
/deploy
```

You can also run PqaDeploy in script mode by specifying a script file with the deploy options. A minimal PqaDeploy script file contains four commands, the name of the PQA file to read, the target hard drive, partition numbers, and the deploy command.

Sample PqaDeploy Script

The following is a sample PqaDeploy script file, NET2K.PQD.

```
// NET2K.PQD

// Replace the network driver with an updated version

// Disable most messages to the screen
Silent

// Record Results to a file on a network drive in case it
// is needed by tech support
Log f:\deploy.log

// Specify the PQA file, which is located on the network
PQA = f:\NET2k.PQA

// Specify the location for performing the update
Drive 1
Partition 1

// Perform the update as specified by the PQA
Deploy

// Now reboot the machine
Reboot

// End of file
```

This script file specifies the name of the PQA file, the target hard drive and partition number, then causes a reboot of the computer upon completion.

To run PqaDeploy in script mode, use the following command:

```
PqaDeploy /cmd=NET2K.PQD
```

This command runs PqaDeploy and reads the script filenamed NET2K.PQD in the current directory. The deploy script command file specifies the deploy options, then executes the commands contained in the specified PQA file. This causes the file net9x.sys in the

\windows\system directory and the net9x.inf file in the \windows\system\inf directory to be deleted. It then writes the new versions of these files contained within the PQA file into the specified locations.

Multiple PQA files may be specified one at a time each followed by a Deploy command in the PqaDeploy script file. This allows many PQA files to be deployed in one execution of PqaDeploy.

RTC Files

A runtime configuration (RTC) file is supplied for each version of PQA. The RTC file has the same name as the corresponding executable, except the extension is .RTC instead of .EXE. Any PqaBuild, PqaDeploy command, or group of commands can be added to the RTC file. These commands are executed each time the application runs. A common command to embed in the RTC file is the PQA file password. This hides the password from customers and protects the PQA file from restoration, except by those with access to the proper PqaDeploy RTC file, which may be BIOS locked as described below.

Any PqaBuild or PqaDeploy command can be disabled in the RTC file. By so doing, the command is not available to customers, thus preventing them from using PqaDeploy in ways not intended. You can also specify an expiration date for PQA in the RTC file.

Both programs also have the ability to “lock” themselves to run only on computers with a specific BIOS or DMI string. BIOS information is specified in the RTC file. When locking is enabled, the system BIOS or DMI is checked before PqaBuild or PqaDeploy is allowed to execute. If the lock test fails, a customizable error message is displayed indicating the failure.

BIOS locking is supported by all versions of both PqaBuild and PqaDeploy.

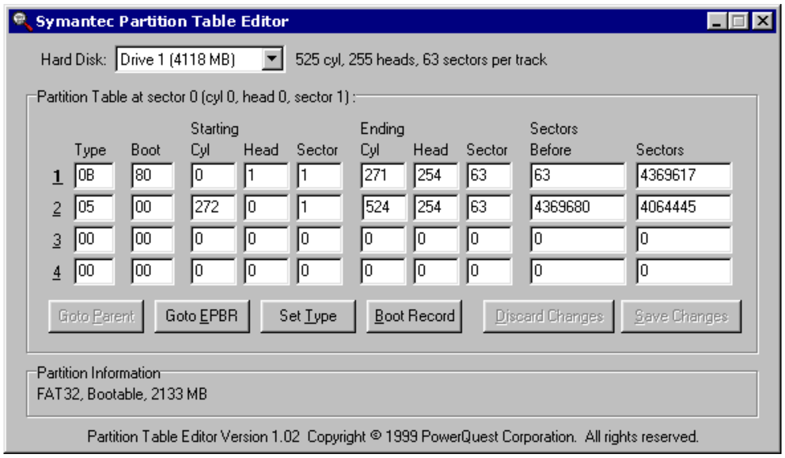
You must contact Symantec Professional Services for RTC file customization.

CHAPTER 6

PTEdit

Overview

Symantec's Partition Table Editor reads and allows manipulation of the partition table information found in the Master Boot Record and EPBR Boot record. PTEdit is useful for fixing partition table errors or boot sector problems. There is a DOS version (PTEdit) and a Windows version (PTEdit32). Both versions have a graphical user interface.



Symantec Partition Table Editor

Hard Disk: Drive 1 (4118 MB) 525 cyl, 255 heads, 63 sectors per track

Partition Table at sector 0 (cyl 0, head 0, sector 1):

			Starting			Ending			Sectors	
	Type	Boot	Cyl	Head	Sector	Cyl	Head	Sector	Before	Sectors
1	0B	80	0	1	1	271	254	63	63	4369617
2	05	00	272	0	1	524	254	63	4369680	4064445
3	00	00	0	0	0	0	0	0	0	0
4	00	00	0	0	0	0	0	0	0	0

Partition Information

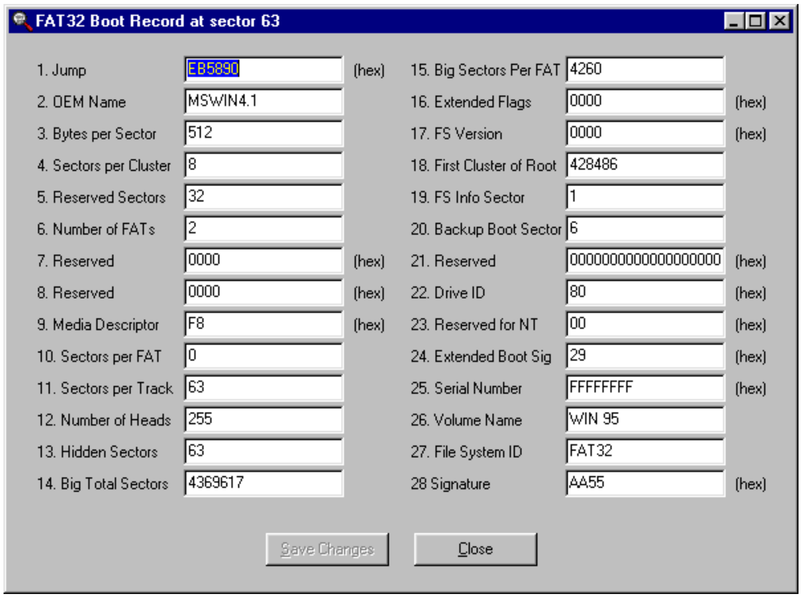
FAT32, Bootable, 2133 MB

Partition Table Editor Version 1.02 Copyright © 1999 PowerQuest Corporation. All rights reserved.

This editor allows you to make changes to partition tables by using decimal values rather than hex values. You can use PTEdit to change the file system flag; set the active partition; hide and unhide partitions; and change CHS values, boot sector information, and the number of sectors in a partition. Keep in mind that when you change the number of sectors, the final result must match the CHS values.

PTEdit looks at partition table information in a relative fashion. Basically it finds the start sector of a primary partition by calculating the absolute value of the sector from the start of the drive to the boot sector of the partition.

The **Boot Record** button is enabled when you select a partition table entry containing either a FAT, FAT32, or NTFS partition. Clicking this button displays a dialog from which you can view or edit the boot record of the specified partition.



FAT32 Boot Record at sector 63

1. Jump	EB5890	(hex)	15. Big Sectors Per FAT	4260	
2. OEM Name	MSWIN4.1		16. Extended Flags	0000	(hex)
3. Bytes per Sector	512		17. FS Version	0000	(hex)
4. Sectors per Cluster	8		18. First Cluster of Root	428486	
5. Reserved Sectors	32		19. FS Info Sector	1	
6. Number of FATs	2		20. Backup Boot Sector	6	
7. Reserved	0000	(hex)	21. Reserved	000000000000000000	(hex)
8. Reserved	0000	(hex)	22. Drive ID	80	(hex)
9. Media Descriptor	F8	(hex)	23. Reserved for NT	00	(hex)
10. Sectors per FAT	0		24. Extended Boot Sig	29	(hex)
11. Sectors per Track	63		25. Serial Number	FFFFFFFF	(hex)
12. Number of Heads	255		26. Volume Name	WIN 95	
13. Hidden Sectors	63		27. File System ID	FAT32	
14. Big Total Sectors	4369617		28 Signature	AA55	(hex)

The boot record dialog displays the contents of the selected partition's boot sector. Each field is numbered, so you can reference fields by number rather than name.

Fixing Partition Table Errors

1. Run PTEDIT.
2. *(If necessary)* Select a different hard drive from the **Hard Disk** drop-down list.
3. Locate the partition you want to change. (If the error occurred on a logical partition, click **GoToEPBR** to locate the correct partition.)

   You may want to record the settings before you make changes, in case the conversion does not work as planned and you want to reset the values afterward.
4. Click in the **Type** field for the partition you want to change, then click **Set Type**.

   A list of available partition types will be displayed.
5. Select the appropriate partition type (for example, FAT32 if the partition type was FAT32 before the problem occurred).

   When a field is edited, the changes are indicated in red and the **Save Changes** button becomes enabled.
6. Click **Save Changes** to write the changes to disk (you will be prompted to confirm this action), or click **Close** to discard the changes and return to the partition table dialog.

Buttons

GOTO Parent button

This button is only enabled when the selected partition table entry represents an EPBR. Clicking this button returns PTEDIT/PTEDIT32 to the parent EPBR/MBR of the currently displayed EPBR. Each screen represents an MBR or EPBR. (The logical is just an entry in the EPBR screen.)

GOTO EPBR button

This button is only enabled when the partition table contains an extended or EPBR partition table entry (type 05 or 0F). Clicking this button causes PTEDIT/PTEDIT32 to read the EPBR and display its partition table. You can also access the EPBR in PTEDIT32 by double-clicking any dimmed button. The **MBR Sector Selection** dialog appears.

You can enter the start sector for that partition/EPBR and click OK. The main screen will then show the information in that sector. This is useful for re-creating the EPBR/logical partitions.

Set Type button

This button is enabled when any of the partition table entry rows have been selected. Clicking this button displays a dialog from which you can select an available partition type.

Boot Record button

This button is enabled when you select a partition table entry containing either a FAT, FAT32, or NTFS partition. Clicking this button displays a dialog from which you can view or edit the boot record of the specified partition.

The boot record dialog displays the contents of the selected partition's boot sector. When a field is edited, the changes are indicated in red, and the **Save Changes** button becomes enabled.

Discard Changes button

As you change a value in any of the partition table fields, the change is indicated in red and this button becomes enabled. Clicking this button discards your changes and re-reads the sector's original information from disk.

Save Changes button

As you change a value in any of the partition table fields, the change is indicated in red and this button becomes enabled. Clicking this button saves your changes to disk. Before the change is made permanent, the program asks you whether you want to continue with the change.

CHAPTER 7

Virtual Boot Environment (VBE)

Overview

The Symantec Virtual Boot Environment (VBE) allows non-Windows programs to execute from off of the hard drive on Windows 9x, Windows Me, Windows NT, Windows 2000, Windows XP, and Linux platforms as if they were running from a boot floppy or from a bootable CD.

The Symantec VBE operates by temporarily replacing the Master Boot Record (MBR) boot code on the primary hard drive. Upon rebooting the computer, the new boot code loads and runs the programs stored in the VBE image file. This Virtual Boot Environment consists of a loader program, disk I/O redirector TSR program, and a non-compressed floppy boot image file. The boot image file is created from a 1.44 MB or 2.88 MB format bootable floppy. It contains both the bootable OS files and the user application programs and data.

Upon booting, the VBE loader builds a RAM DRIVE in memory the size of the boot image file, copies the boot image file from the hard drive to memory and then loads and executes the boot sector within the boot image file. Under DOS, the A: drive is then used to reference the boot image file system just like when booting from a CD. The first physical floppy drive is accessed as drive letter B:.

Programs and data can be both read and written to the VBE A: drive. However, any changes written to the A: drive will be lost when the computer is rebooted or the power is cycled, just like when using a standard RAM DRIVE.

The FAT and FAT32 partitions defined on the computers hard drives are accessible as drive letters C:, D:, E:, etc. NTFS, Ext2, Ext3, and other partition types can be accessed using Symantec programs like PQAccess and PQA.

The VBE image file is a raw sector-by-sector representation of a floppy file system in either a 1.44 MB or a 2.88 MB floppy disk format. It contains everything that would exist on a bootable floppy disk (boot sector, FATs, directories, and files.)

Using the Virtual Boot Environment

There are four steps required to use the Symantec Virtual Boot Environment.

1. Build the boot image file, PQVF.VFD.
2. Building the VBE directory on the hard drive.
3. Run the Virtual Boot Environment Enabler program.
4. Reboot the computer to execute the VBE loader.

Each of these steps are detailed in this chapter.

Several VBE image files can be created and stored in the VBE directory with descriptive image filenames like w98boot.ima, w2kboot.ima, w95bboot.ima, etc. Batch files can then be created that will copy or rename each image file to PQVF.VFD before executing the VBE installer. Under a GUI environment, icons can be created that will boot to the image needed with just a click of the mouse.

Building the Boot Image File

The VBE image file is a raw sector-by-sector representation of a floppy file system in either a 1.44 MB or a 2.88 MB floppy disk format. It contains everything that would exist on a bootable floppy disk (boot sector, FATs, directories, and files.)

Using a Physical Boot Floppy

One way to build the boot image file is to build an actual physical bootable floppy disk. Place all needed files including the operating system files, drivers and application programs and data files onto an actual floppy. The created boot floppy can be easily tested for correct operation by booting the floppy on the target computer. Changes can be quickly made and tested.

When the boot floppy operates correctly, you can use utilities like VFILE (or VF Editor, which is included with Symantec DeployCenter) to read the sector-by-sector data from the floppy into an image filenamed PQVF.VFD. Make sure that you disable any image compression options. You can check for compression by examining the image file size to make sure it is either 1.44 MB (1,474,560) or 2.88 MB (2,949,120).

After the desired VBE boot image file is created, the next step is to run the VBE installer.

Using a Boot Image File

Another way to build the boot image file is to use the VFILE program. You can use VFILE to create and manage the bootable floppy image file directly on the computer's hard disk without first creating a physical floppy disk.

VFILE is a command line program that lets you create and manage floppy image files. Two versions are supplied, one for Windows (VFILE) and one for DOS (VFILEX). VFILE can be scripted to automate the creation or modification of the image file and it's contents. For more information about VFILE, see "VFILE System Utility" on page 83.

If you have Symantec DeployCenter, you can also use Symantec VF Editor to create and manage floppy image files in a GUI environment. You can use drag and drop file operations to add, delete, and replace files in the image file. You can read or write a physical floppy disk to or from an image file. See the help file included with the program for more information.

Additional Notes

When you build the boot image file, there are issues with the following DOS drivers: HIMEM.SYS, EMM386.EXE, and VDISK.EXE.

- If any extended memory or the UMA area is going to be used, HIMEM.SYS is required to be loaded by the CONFIG.SYS file. For some versions of DOS, this functionality may be included in the EMM386.EXE program.
- The MS-DOS 6.x versions of HIMEM require the HIMEM /TESTMEM:OFF command line switch to be specified to prevent HIMEM from testing memory and thus overwriting the VBE image in memory.

You cannot use the VDISK.EXE RAM disk with the VBE. Use the newer RAMDISK program to provide a RAM disk. The RAMDISK program uses a memory allocation system that works with HIMEM.SYS whereas VDISK does not.

Building the VBE Directory

1. Create a subdirectory on the primary boot drive of the computer to contain all the VBE files. Copy the required VBE files VFLOPPY.SYS, VFLOPPYLD.SYS, VFRSTMBR.COM, RESTRMBR.EXE, and REBOOT.COM to the new subdirectory.
2. Copy the appropriate VBE installer files for the operating system that will be running when the VBE enabler program will be executed.
3. Name the virtual floppy image file PQVF.VFD, and copy it into the subdirectory.

With these files present in the subdirectory, you can run the enabler program, reboot the computer and run the VBE to execute the programs and batch files stored within the virtual floppy image.

VBE Enabler Program

The VBE enabler program prepares the computer to boot to the VBE during the next computer boot up sequence. There are three versions of the VBE enabler program:

- VFINSTNT.EXE – Windows NT, Windows 2000, and Windows XP.
- VFINST9X.EXE – Windows 9x, Windows Me.

- VFINSTD.EXE – DOS

The VBE installer saves the current MBR to a backup file (MBR.DAT), writes the new MBR boot code to sector 1 and to two other sectors on track one of the hard drive, records in the VBE loader the physical disk location of the VBE image file, PQVF.VFD, located in the current directory, and then reboots the computer.

The VBE enabler program recognizes several optional arguments.

- /AUTORMV = [ON|OFF]- If neither is specified or if this switch is ON the original MBR boot code is restored after the first VBE boot sequence. This allows for a single boot to the VBE, and then automatically restores the normal boot sequence. If you need to boot to the VBE several times, specify /AUTORMV=OFF.
- /DBG=DBG.TXT - Creates a debug output file.
- /NRB - Causes the VBE enabler program to not reboot after completion.

The enabler program takes several seconds to run (watch for the disk activity to subside), generates no output to the display, and returns an error code on failure to the command processor ERRORLEVEL variable. This error code can be tested using the IF batch command.

VBE Installer Error Return Values

1	Internal Error
2	Engine Error
3	Couldn’t find partition data structure for filename
4	The file system on the drive you are installing Virtual Floppy on is corrupt
5	A file operation failed on VFLOPPY.SYS
6	A file operation failed on VFLPPYLD.SYS
7	Cannot get the sectors for PQVF.VFD
8	Cannot get the sectors for VFLOPPY.SYS
9	Error Writing to Disk
10	Error reading from disk
11	Couldn't write MBR.DAT or write MBR to driver file
12	Not enough unused sectors available on first track to install virtual floppy
13	Backup Head Fail
14	Restore Head Fail
15	Cannot write to VFLOPPY.SYS or VFLPPYLD.SYS
16	Cannot read VFLOPPY.SYS or VFLPPYLD.SYS
17	Drive overlay software prevents install of Virtual Floppy
18	VFLOPPY.SYS version is not supported by installation
19	VFLPPYLD.SYS version is not supported by installation
20	VFLOPPY.SYS is corrupt
21	Cannot read PQVF.VFD file
22	The PQVF.VFD file is not the size of a valid floppy image

Diagnosing VBE Enabler Problems

You can use the /DBG switch to create a log file that contains information to help solve enabler problems. For example, to run the VBE installer under Windows 9x, use the following command:

```
VFINST9X /AUTORMV /DBG=OUTPUT.LOG
```

This will setup the VBE to boot the PQVF.VFD boot image file located in the current directory. The optional arguments specify to automatically restore the normal boot sequence after the first VBE boot and to write a debug log to the OUTPUT.LOG file. If the enabling is successful, the computer will be automatically rebooted into the VBE environment.

Running the Virtual Boot Environment

When the VBE loader has been enabled, the next time the computer is booted the VBE loader receives control, loads the PQVF.VFD file, and executes the boot programs contained in the floppy image. During the loading process the following characters are displayed on the screen to indicate the loader's progress.

```
|....
Msg 11  Loading Virtual Floppy
Msg 12  Virtual Floppy Loaded
```

At this point the virtual floppy, which has now been loaded into memory, should run and display a DOS load message like "Starting Caldera DR DOS" or "Starting Windows 98..." or "Starting Windows Millennium Emergency Boot".

VBE Loader Error Messages

In the case of an error, the loader outputs one or more of the following error messages.

Err 10: Error reading sector
Err 13: Attempt to read past cyl 1024
Err 14: Automatic removal of Fake Floppy failed
MSG: 11 Loading floppy image
MSG: 12 Floppy image loaded

Upon a successful boot, the VBE loader restores the original MBR back to sector 1 unless the /AUTORMV=OFF switch was specified when the VBE installer was run. This will cause the normal hard drive boot sequence to be activated the next time the computer is booted.

If you need to boot back into the VBE a second time, you must specify the /AUTORMV=OFF switch when running the VBE installer. When you are ready to switch back to the normal hard drive boot sequence you can then use either the RESTRMBR or VFRSTMBR programs to restore the original MBR.

Holding down the right shift key, while booting the computer, will cause the loader to skip the VBE boot sequence and perform a normal hard drive boot.

Restoring the Original Boot Sequence

The RESTRMBR and the VFRSTMBR programs both restore the MBR boot code. The smaller VFRSTMBR program restores to MBR saved by the VBE enabler program and may only be used while the computer is booted into the VBE.

The larger RESTRMBR program can be used to restore either generic or alternate MBR boot code. It takes an optional full path name to the file that contains the saved boot sector. For example:

```
RESTRMBR C:\TMP\MBR.DAT
```

To restore generic MBR boot code, run RESTRMBR without an argument. This functions just like FDISK /MBR. This may cause unexpected results if you have programs installed that contain their own version of the MBR boot code, such as Symantec's BootMagic.

After restoring the MBR boot code, the REBOOT program can be used to reboot the computer.

Distribution File List

The following files are included in the Symantec Virtual Boot Environment.

- VFINSTNT.EXE - VBE Installer for Windows NT/2000.
- VFINST9X.EXE - VBE Installer for Windows 9x/Me.
- PQVXD.VXD - Windows Driver for Windows 9x/Me Installer.
- VFINSTD.EXE - VBE Installer for DOS.
- VFLOPPY.SYS - VBE First phase loader.
- VFLPPYLD.SYS - VBE Second phase loader and disk I/O redirector.
- RESTRMBR.EXE - Restore old or alternate MBR from a file or generic version.
- VFRSTMBR.COM - MBR restore program, calls TSR to do restore.
- REBOOT.COM - Reboot program for DOS.
- VFILE.EXE - Boot image file creation and manipulation utility for Windows.
- VFILEX.EXE - Boot image file creation and manipulation utility for DOS.
- SAMPLE.VFD - Sample boot image file with Caldera DOS OS files.
- CALDERA.MBR - Caldera DOS boot sector image file.

CHAPTER 8

VFILE System Utility

The VFILE program can be used to create and modify a bootable floppy image file directly on the computer without needing to first create a physical floppy disk. VFILE can be run in either command line or batch mode.

In command line mode, it prompts you for commands to be entered from the console one at a time. In batch mode, it processes a batch file that contains one or more VFILE commands. Each command is listed one per line. To run VFILE in batch mode, use the CALL command and specify the batch filename. An example of using VFILE to create a bootable floppy image file is listed at the end of this section.

There are two versions of VFILE: one for use in a Windows DOS box (VFILE) and one for use in DOS (VFILEX). The calling syntax is as follows:

```
   Vfile[X] [-<Switch>...] [<Command> [-<h|?> | <Parameters...>]]
```

Commands shown in square brackets [ ] are optional.

Switches

-f<FileName>	Mount image file. Repeat for additional images.
-h,-?	Show help message.
-e	Environment. Load internal environment from system.

Environment variables can be manipulated with SET command.

Type `VFile Help` to view a list of all the available commands.

Type `VFile <Command> -h` to view help information on each command.

If no command is specified, console mode is entered. If no image file is mounted and if it exists in the current directory, a filenamed IMAGE.IMG is mounted.

VFILE Command List

Call
CheckMedia
CompFile
CopyFile
CreateFile
DelDir
DelFile
Dir
Dup
Echo
Erase
Exit
Format
Goto
Help
If
ListDrives
MakeDir
Mount
NewImage
Pause
Rem
Set
Tree
Type
VerifyFile

Volume and Image File Related Commands

The volume and image file related commands are Mount, Format, NewImage, ListDrives, and CheckMedia.

Mount

The mount command is used to select a virtual image file. Mounted images use the "\\.\VFn" filename format as volume names. The volume number increments for each virtual image file mounted. Once a virtual image file is mounted, any of the file commands may reference the image using the \\.\VFn\filename format.

Syntax: Mount <ImageFile> [<ImageFile2>...]

Format

The format command is used to format a mounted virtual volume and to write the boot sector. By default a Windows 98 type boot sector is written. To use other boot sector types, specify the optional boot sector image filename.

Syntax: Format [-4] [-8] [-2] [-s] <Drive> [BootImage]

-4 = 1.44M floppy.
-8 = 2.88M floppy.
-2 = 1.2M floppy.
-s = Default boot sector. [BootImage] will be ignored if specified.

[BootImage] = Name of an Image file containing the boot sector to use.

Examples:

```
Format \\.\VF0
```

```
Format 0
```

NewImage

The NewImage command creates a new empty image file. You specify the filename of the new image file, the size of the image and the optional boot sector image filename.

Syntax: NewImage [-4] [-8] [-2] [-s] <Filename> [Sectors] [BootImage]

-4 = 1.44M floppy image.
-8 = 2.88M floppy image.
-2 = 1.2M floppy image.
-s = Write default boot loader.

<Filename> = Image filename to create.

[Sectors] = Size of image file in number of 512 byte sectors. If an image type switch is specified a file will be created that fills the media. This parameter is required if the BootImage parameter is specified.

[BootImage] = Name of an Image file containing boot sector to load. Ignored if -s is specified.

ListDrives

The ListDrives command lists the mounted Virtual File System drives and information.

Syntax: ListDrives [-a]

-a = Show all visible drives, not just mounted virtual volumes.

CheckMedia

The CheckMedia command reports the current media status.

Syntax: CheckMedia [Drive]

Examples:

```
CheckMedia \\.\Vf0
CheckMedia 0
CheckMedia C:
CheckMedia
```

File System Access Related Commands

The file system access related commands are Dir, .Copyfile, DelFile, Dup, Erase, Type, CreateFile, VerifyFile, Compfile, MakeDir, and DelDir.

Dir

The DIR command lists the files defined in the mounted volume. Standard DOS format wildcards are allowed.

Syntax: Dir [[[Path]\][File(s)]]

CopyFile

The CopyFile command copies a single file.

Syntax: CopyFile [-d] <SourceFile> <DestinationFile>

-d = Delay for 5 seconds every 50MB.

DelFile

The DelFile command deletes a single file.

Syntax: DelFile [-u] <FileName>

-u = Unconditionally delete a file with the read-only attribute set.

Dup

The Dup command duplicates or copies file(s).

Syntax: Dup [-p] [-s] [-v] [-d] <File(s)> [DestPath]

-p = Display progress.
-s = Show status/statistics.
-v = Verbose mode. Show filenames as they are copied.
-d = Delay for 5 seconds every 50MB.

Examples: `Dup *.Txt \\.\Vf0` (Copy from current directory to target root.)

`Dup \\.\Vf0\*.Dat` (Copy from drive 0 root to current directory.)

Erase

The Erase command deletes one or more files.

Syntax: Erase [-u] [-q] <File(s)>

-u = Unconditionally delete files with the read-only attribute set.
-q = Quiet mode. Don't show filenames as they are deleted.

Example: `Erase \\.\Vf0\*.Txt` (Erase all text files in drive 0 root.)

Type

The Type command displays the contents of a file.

Syntax: Type <Filename>

CreateFile

The CreateFile command creates a new file using a predictable data pattern or template file.

Syntax: CreateFile [-d] [-p] <FileName> [Size] [Template]

-d = Delay for 5 seconds every 50MB.
-p = Pattern. Create with predictable data pattern.
[Size] = File size in bytes. *(If omitted, file is created that fills the media.)*
[Template] = Template file to duplicate at start of created file.

VerifyFile

The VerifyFile command verifies the contents of a file by reading all bytes in file.

Syntax: VerifyFile [-r] [-c] <FileName>

-c = Check for data pattern created by CreateFile command.
-r = Process file in reverse.

CompFile

The CompFile command compares two files.

Syntax: CompFile [-r] <File1> <File2>

-r = Process files in reverse.

MakeDir

The MakeDir command creates a new subdirectory.

Syntax: MakeDir <DirName>

DelDir

The DelDir command deletes a subdirectory.

Syntax: DelDir <DirName>

Console Environment and Batch Processing Related Commands

The console environment and batch processing related commands are Help, Goto, If, Set, Call, Exit, Echo, Rem, and Pause.

Help

The Help command lists either the available commands or additional help on an individual command.

Syntax: Help [Command]

Goto

The Goto command branches to a new position in the batch file.

Syntax: Goto <Label>

Example: `Goto LABEL` (Branch execution to line beginning ':LABEL'.)

If

The If command executes a command if expression-1 is the same as expression-2.

The comparison is case insensitive for strings.

Syntax: If <Expr1> <== or !=> <Expr2> <Command> [Command Args...]

Set

The Set command assigns a value to an environment variable.

Syntax: Set <Variable>=[Value]

Call

The Call command executes VFILE commands from a batch file.

Syntax: Call <BatchFile> [Args...]

Exit

The Exit command ends processing of a batch file.

Echo

The Echo command displays text on the console or controls the echoing of batch file lines.

Syntax: Echo [On | Off | <Text>]

Example:

`Echo ON` = Turns on echo.
`Echo OFF` = Turns off echo.
`Echo <text>` = Displays text.
`Echo` = Displays echo state.

Rem

The Rem command allows remarks to be placed into batch files.

Pause

The Pause command displays the optional text and then stops processing till a return key is pressed on the console.

Syntax: Pause [Text]

Example VFILE Batch Command File

This example VFILE batch command file creates a bootable Caldera DOS virtual image file with the PQBOOT.EXE and DISE.EXE programs. It turns off the command echo, creates a new 1.44 floppy image filenamed TEST.VFD, writes a Caldera DOS boot sector, mounts the new image file, copies files to the image, and then lists all the files in the image. If this batch file were named EXAMPLE.CMD it would be executed using the following command:

```
C:> VFILE CALL EXAMPLE.CMD
```

Contents of EXAMPLE.CMD:

```
echo off
```

```
rem Example VFILE batch command file
rem to create a Caldera DOS boot image
newimage -4 test.vfd 2880 calderab.bin
mount test.vfd
copyfile IBMBIO.COM   \\.\vf0\IBMBIO.COM
copyfile IBMDOS.COM   \\.\vf0\IBMDOS.COM
copyfile COMMAND.COM  \\.\vf0\COMMAND.COM
copyfile MOUSE.COM    \\.\vf0\MOUSE.COM
copyfile PQBOOT.EXE   \\.\vf0\PQBOOT.EXE
copyfile DISE.EXE     \\.\vf0\DISE.EXE
copyfile DISE.RTC     \\.\vf0\DISE.RTC
copyfile HIMEM.SYS    \\.\vf0\HIMEM.SYS
copyfile CONFIG.SYS   \\.\vf0\CONFIG.SYS
copyfile AUTOEXEC.BAT \\.\vf0\AUTOEXEC.BAT
copyfile RUNDISE.BAT  \\.\vf0\RUNDISE.BAT
copyfile SCRIPT.TXT   \\.\vf0\SCRIPT.TXT
dir \\.\vf0
exit
```

The resulting image file, TEST.VFD, contains a bootable Caldera DOS operating system with PQBOOT, DISE, batch and script files.

Copying a Floppy Disk to an Image File

Use the following steps to copy a bootable floppy disk to an image file using the VFILE program.

1. Run the **VFILE** (or VFILEX) program.
2. At the **VFILE>** prompt, type ”`newimage -8 -s <Filename>`” then press **Enter**. The response “2,949,120 bytes were written” is displayed.

   <Filename> can be any legal filename, however, something descriptive like “WIN98DOS.IMA” or “DOS622.IMA” makes it easier to later identify what the image file contains.
3. At the **VFILE>** prompt, type ”`mount <Filename>`”, using the filename from step #2 above. Press **Enter**. The response “<filename> Mounted as Volume \\.\vf0” is displayed.
4. Insert a bootable floppy disk into the A: drive.
5. At the **VFILE>** prompt, type `dup  -v  A:\*.*  \\.\vf0\` Press **Enter**. All files are copied from the root of A:\ to the image file mounted as \\.\vf0. Copying files from subdirectories of A: or any other source directory requires additional commands.
6. Type ”`exit`” and press **Enter**. You are returned to a DOS command prompt.
7. Rename the image file created to the filename PQVF.VFD. Place the PQVF.VFD file in the same directory as the VBE installer program.

CHAPTER 9

PQDisk

Overview

PQDisk is a scriptable version of Norton™ PartitionMagic®. It is intended for use by PC configuration centers and corporations that need to configure large numbers of PCs. Unlike Norton PartitionMagic, PQDisk has no interactive user interface. It uses ASCII text script files based on the Norton PartitionMagic Pro scripting language to specify operations that create and manipulate the computer's hard disk partitions. PQDisk provides optional log and error files and can display script execution progress status. The log file details the operation of each script command. Error files contain only the error number so it can be easily parsed for automated processes.

You can use PQDisk to create, format, copy, move, resize and delete partitions. Additional operations are also available to display information about partitions, test bad sectors, analyze clusters, convert file system types, and change a partition's status (active, hidden).

To allow better control over PQDisk's behavior, many of the constraints built into the retail version of Norton PartitionMagic are disabled in PQDisk. For example when a primary partition is hidden or unhidden with PQDisk, other primary partitions are not affected, whereas if you set one primary partition active with Norton PartitionMagic, all other primary partitions on that hard disk are hidden.

PQDisk uses a three-step process to perform partition operations. The first step selects the physical disk where the partition is located, the second step is to select a free space or defined partition, and the third is to select the operation. You can also specify various options for each operation.

PQDisk functions on computer systems with hardware-based RAID systems. It will not function on software-based RAID systems.

PQDisk System Requirements

Hardware/ Software	Windows 9x, Windows Me, and MS-DOS	Windows NT, Windows 2000, Windows XP
Processor	Intel 486 DX or above (33MHz)	Intel 486 DX or above (33MHz)
RAM	16 MB (32MB recommended for FAT32 partitions)	16 MB (32 MB recommended for FAT32 partitions)
Hard disk space	1 MB	1 MB

*More memory may be required to manipulate FAT32 partitions on hard disks larger than 4 GB.

Supported File Systems

PQDisk supports the following partition types and file systems.

File System	Description
Extended	The extended partition gets around the arbitrary four-partition limit for a disk. An extended partition is a container in which you can further divide your disk space by creating logical partitions. An extended partition does not directly hold data. You must create logical partitions within the extended partition to store data.
Extended-x	An extended-x partition functions like an extended partition but is not limited to the first 1024 cylinders on a drive. Linux kernels below 2.2 do not support Extended-X partitions.
FAT	Uses file allocation table (FAT) and clusters. The FAT file system is used by DOS, Windows 3.x, and most Windows 9x and Windows Me installations. A FAT partition is also accessible by Windows NT/2000/XP and OS/2.
FAT16x	FAT16x is a proprietary file system developed by Microsoft to enable FAt partitions beyond 1024 cylinders (~8GB).
FAT32	FAT32 is an enhancement of the FAT file system. Because it uses 32-bit file allocation table entries, rather than the 16-bit entries used by the FAT system, FAT32 supports larger disk or partition sizes (up to 2 terabytes). The minimum size for a FAT32 partition is 256 MB. A FAT32 partition is accessible by Windows 95 OSR2 (version 4.00.950B), current versions of DOS, Windows 98, Windows Me, Windows 2000 and Windows XP. Windows 3.x, Windows NT 3.51/4.0, earlier versions of Windows 95 and early versions of DOS (MS-DOS 6.22, PC-DOS 7.0, and DR-DOS 7.0 or earlier) do not recognize FAT32 and cannot use files on a FAT32 partition.

File System	Description
FAT32x	FAT32x is a proprietary file system developed by Microsoft to enable FAT32 partitions beyond 1024 cylinders (~8GB). Windows 95 OSR2 and Windows 98 may use FAT32x partitions.
Linux Ext2 and Ext3	The Linux Ext2 and Ext3 file systems are only accessible by Linux, a freeware version of UNIX. The Linux Ext2 and Ext3 file systems support a maximum partition size of 4 terabytes.
Linux swap	Holds a Linux swap file. The maximum usable size of a Linux swap file is 128 MB. The default size shown when you create a Linux swap partition may be slightly larger because of the physical geometry on the hard disk.
NTFS	The New Technology File System (NTFS) is accessible only by Windows NT, Windows 2000, and Windows XP. NTFS is not recommended for use on disks less than 400 MB because it uses a great deal of space for system structures.
Unformatted	Unformatted partitions reserve a portion of the disk but are not assigned a file structure.
Unallocated	Unallocated space is the portion of a hard disk that is not currently assigned to any partition. Sometimes called free space.

Integrity Checks

PQDisk checks disk integrity with a sophisticated system of analysis and validation that operates behind the scenes every time you complete an operation.

If your physical disk fails the integrity check, the Select Drive command (see page 121) will not work correctly. This is because of a problem with the disk, not because of a problem with PQDisk.

You can use the Set Partition Table Autofix command (see page 122) to allow PQDisk to find and fix errors automatically. It is safe to allow PQDisk to fix errors. Correct the disk problem, and then restart PQDisk. See “Resolving Partition Table Errors” on page 199.

In addition to the integrity check at startup time, PQDisk performs two integrity checks during any operation. The first check tests the integrity of the file system in the partition before an operation begins (similar to CHKDSK or ScanDisk), and the second check validates your disk’s data after an operation is completed. From start to finish, PQDisk examines your disk and informs you immediately if it detects any irregularities.

PQDisk Operation Preferences

PQDisk has several preference settings that affect its behavior.

- FAT partition 64K FAT Clusters
- Bad Sector Checking
- Drive Read-Only Protection

These preference settings need to be correctly set for your requirements to insure proper operation.

Allow 64K FAT Clusters for Windows NT

This preference lets you create FAT partitions with 64 KB clusters, which enables Windows NT, 2000, XP to support large hard disks.

IMPORTANT! You cannot access a 64K partition from DOS, Windows 3.x, Windows 9x, or Windows Me. You should only use 64K partitions with Windows NT. If you are using multiple operating systems, we recommend not using 64K clusters.

To prevent you from inadvertently creating partitions with 64K clusters, this preference is disabled every time you start PQDisk. When enabled, the 64K cluster size is available in the Resize, Move Partition, and Resize Clusters commands.

Skip Bad Sector Checks

When PQdisk modifies partitions, it performs extensive testing to detect bad sectors on your hard disk. Newer disk types (such as Enhanced IDE and SCSI) often handle bad sectors internally, making such testing superfluous. For this reason, PQDisk lets you bypass these tests with **Skip Bad Sector Check**s. When this preference is enabled, the Resize, Move, Create, Copy, and Format operations run faster.

WARNING! If you skip bad sector checks and your hard disk has bad sectors, data loss can result.

Bad sector checking is off by default. PQDisk lets you set this preference individually for each of the hard disks. A check mark next to a disk means to skip bad sector checking for that disk.

Set a Disk as Read-Only

This preference lets you prevent PQDisk from making any changes to a hard disk. You can set this preference individually for each of the hard disks. There are some exceptions to how this preference is applied: If the disk contains the boot partition, some files may be changed, such as the Windows NT/2000/XP boot initialization (BOOT.INI) file.

Operations

PQDisk allows partitions to be created, formatted, copied, moved, resized and deleted. Additional operations are also available like: partition information display, bad sector testing, cluster analysis, file system type conversions, and partition status changes (active and hidden).

Creating Partitions

The Create operation lets you create primary partitions, extended partitions, and logical partitions. On a single hard disk, you can have up to four primary partitions or three primary partitions and one extended partition. Within an extended partition, you can create unlimited additional subdivisions called logical partitions.

Generally, you should create primary partitions to install operating systems and logical partitions for all other purposes, such as storing data and applications. If you have multiple hard disks, you can improve speed by installing operating systems and applications on separate disks.

To create a partition, free space must exist on the hard disk. If no unallocated space exists, you must resize or delete an existing partition to create unallocated space.

Managing Drive Letter Changes

Creating a new partition may cause your drive letters to change. For example, if you have one primary partition (C:) on your hard drive and a CD drive (D:), and you create a new logical partition on your hard drive, the new partition becomes D: and the CD drive changes to E: after you reboot your computer. As a result, any programs on your hard drive that were linked to the CD no longer function because the paths to files have changed.

Shared Privileges

If you have set up shared privileges under Windows 9x or Windows Me, you should check them after creating or deleting partitions. If your drive letters change, you could inadvertently lose shared privileges or share private information.

Creating Bootable Partitions

Before creating a partition where you plan to install an operating system (a bootable partition), you should understand the following information.

Operating System	Boots from	Supported Partition Types	Boot Code Boundary	Space Required
DOS 6.22 and earlier	Primary	FAT	2 GB	8 MB
Windows 95a	Primary	FAT	2 GB	90 MB
Windows 95b	Primary	FAT or FAT32	8 GB	90 MB
Windows 98	Primary	FAT or FAT32	8 GB	175 MB
Windows 98SE	Primary	FAT or FAT32	8 GB**	190 MB
Windows Me	Primary	FAT or FAT32	8 GB**	300 MB
Windows NT	Primary*	FAT or NTFS	2 GB	120 MB
Windows 2000	Primary*	FAT, FAT32, or NTFS	8 GB**	650 MB
Windows XP	Primary*	FAT, FAT32, or NTFS	8 GB**	>1 GB
Linux (LILO***)	Either	Linux Ext2/Ext3**** and Linux Swap	8 GB	>250 MB

* Windows NT/2000/XP must boot from a primary partition on the first drive. However, only a few files must reside on that partition; the remaining files can reside on a logical partition, which can be located on the first or a subsequent drive. The Windows NT/2000/XP boot partition can be shared with another operating system.

** Having an LBA-compatible (Logical Block Addressing) MBR (Master Boot Record) will make the boot code boundary null with Windows Me/2000/XP.

*** If you install LILO to a logical partition, it must be the first logical partition in the extended partition.

**** Linux also supports the partition types FAT, FAT32, and NTFS (read-only) if Linux is installed to a Linux Ext2 or Ext3 partition.

IMPORTANT! When you create, move, or resize a bootable partition, the partition must begin below the boot code boundary specified in the above table in order for the operating system to boot. With the exception of DOS 6.22 (or earlier), partitions beyond 8 GB are visible to the current operating system.

PQDisk generates a warning message in the log file if you attempt to create, move, or resize a bootable partition outside of the 2 GB boot code boundary. If you continue with the operation, you may not be able to boot or to see the partition. In either case, you can resolve the problem by moving the partition back within the boot code boundary.

If your system includes SCSI disks and you create a partition before a bootable Linux partition, Linux may no longer be bootable. In this situation, you may need to create Linux rescue disks, boot from the rescue disks, and repair the Linux boot information on the Linux partition.

* Windows NT must boot from a primary partition on the first drive. However, only a few NT files must reside on that partition; the remaining files can reside on a logical partition, which can be located on the first or a subsequent drive. The NT boot partition can be shared with another operating system.

**FAT32 is only supported by Windows 95 OEM Service Release 2, Windows 9x, Windows Me, Windows 2000, and Windows XP.

Deleting Partitions

The Delete operation deletes a partition and makes its data inaccessible. To delete an extended partition, you must first delete all logical partitions within the extended partition.

Deleting a partition can make your drive letters change, causing applications not to run because application shortcuts, initialization files, and registry entries refer to incorrect drives.

If your system includes SCSI disks and you delete a partition before a bootable Linux partition, Linux may no longer be bootable. In this situation, you may need to create Linux rescue disks, boot from the rescue disks, and repair the Linux boot information on the Linux partition.

Changing Partition Labels

The Label operation lets you display and change a partition's volume label. Giving your partitions meaningful names makes managing them easier. Labels can be up to 11 alphanumeric characters for FAT partitions, 16 characters for Ext2 or Ext3 partitions and 255 characters for NTFS partitions. Labels follow the same rules as DOS names, with two exceptions: spaces are allowed, and no period is required between the first eight characters and the last three. Labels cannot contain the following special characters: [*?:<>|+=;\/",].

Formatting Partitions

The Format operation formats a partition, destroying all its data in the process.

PQDisk also has several conversion options that let you convert from one file system to another without destroying existing files in a partition.

Copying Partitions

The Copy operation lets you to make an exact duplicate of a partition. Reasons why you might want to copy a partition include:

- To duplicate your operating system before upgrading to a new version or a different operating system (so that you can remember how the old operating system's windows, program icons, and properties were set up).
- To quickly move a smaller hard disk's contents to a larger, new hard disk.
- To change the relative order of partitions.
- To back up a partition.

The copy is the same size (or slightly different if copied to a disk with a different geometry) and file system type and contains the same data as the original.

Checking Partitions

The Check operation checks the integrity of a partition.

PQDisk can only check partitions that it can lock (that is, partitions that do not have open files on them). If there are open files on a partition, the Check option will fail and an error message will be written to the log file.

Check does not display information about the status and structure of a partition as do the DOS, Windows, and OS/2 CHKDSK utilities. To view that information, use the Info command (see page 112).

Getting Information About Partitions

The Info operation displays information about the status and structure of a selected partition. The following option pages are available:

- Disk Usage
- Cluster Waste
- Partition Info
- File System

Disk Usage

The Disk Usage page is available for the FAT, FAT32, and NTFS file systems. This page displays the following information in bytes, megabytes, and as a percentage:

- Used space on the partition, including space wasted by clusters
- Unused space on the partition
- Bad space on the partition
- Total space on the partition (the sum of Used, Unused, and Bad space)

Cluster Waste

The Cluster Waste page applies only to partitions using the FAT or FAT32 file systems. This page displays the following information:

- Current Cluster Size in bytes or kilobytes
- Data stored on the partition in bytes and megabytes

- Wasted space on the partition in bytes and megabytes
- Total used space in bytes and megabytes (the sum of Data and Wasted space)

Partition Info

The Partition Info page is available for all types of partitions, including unallocated space and extended partitions. Information on this page includes the following:

- Partition type is shown in hexadecimal followed by a text description of the partition or file system type (such as FAT, FAT32, or NTFS). The hexadecimal designation is the conventional way to display partition types.
- Serial Number is shown if the partition's file system uses serial numbers.

The lower portion of the page shows physical information about the partition:

- First physical sector shows the logical number and the location (cylinder, head, and sector) where the partition begins.
- Last physical sector shows the logical number and the location (cylinder, head, and sector) where the partition ends.
- Total physical sectors displays the number of sectors in the partition.
- Physical Geometry shows the total number of cylinders, heads, and sectors on the physical disk where the partition resides.

File System-Specific Info Pages

The last page in the Partition Information dialog corresponds to the file system used on the selected partition. For example, if the file system is FAT or FAT32, the page is FAT Info; if the file system is NTFS, the page is NTFS Info, and so forth.

FAT Info This page applies to partitions using the FAT or FAT32 file systems. The first section provides the following information:

- Sectors per FAT shows the number of sectors in each file allocation table and the number of file allocation tables on the selected partition.
- Root directory capacity shows the number of possible entries and the number of sectors in the root directory. Because a FAT32 root directory can grow as needed, this line is blank for FAT32 partitions.
- First FAT sector shows the logical sector number within the partition where the FAT begins.
- First Data sector shows the logical sector number within the partition where the data portion of the partition begins.

The next section provides the following information:

- The number of bytes in files on the partition, the number of files, and the number of those files that are hidden
- The number of bytes in directories on the partition, the number of directories, and the number of those directories that are hidden

The final section of this page, FAT Extensions, provides the following information:

- The number of bytes used for OS/2 Extended Attributes and the number of files and directories affected by Extended Attributes
- The number of bytes used for long filenames and the number of files and directories using long filenames

NTFS Info This page applies to partitions using the NTFS file system. The first section shows the following information:

- NTFS Version shows the version number. The NTFS version does not match the operating system version. For example, Windows NT 4.0 uses NTFS version 1.3.
- Bytes per NTFS sector displays the number of bytes in each logical sector on the selected partition. (There are always 512 bytes in each physical sector.)
- Cluster size displays the size of each cluster and the number of sectors in each cluster on the selected partition.
- First MFT Cluster shows the logical number of the first cluster in the master file table (MFT).
- File Record Size gives the size of file records in the MFT.

The next section displays information similar to that shown by NT CHKDSK:

- Number of files on the partition and the bytes and clusters allocated to them
- Number of wasted bytes in file clusters
- Number of indexes (directories) and the bytes and clusters allocated to them
- Number of bytes and clusters reserved for other system structures

Scanning a Disk for Errors

The ScanDisk utility included with Windows 95/98/Me scans a partition for errors and fixes them. The Check operation also scans for errors, but it does not correct them.

If you are running Windows NT/2000/XP, you can run CheckDisk, rather than ScanDisk,.

ScanDisk only scans partitions with assigned drive letters; it does not scan hidden partitions, extended partitions, unallocated space, or partitions with file systems not supported by the active operating system.

When Check is finished, the results information about errors on the partition (if any were found) and other disk statistics, such as total disk space, number of bytes in bad sectors, and total allocation units are written to the log file.

Retesting Bad Sectors

The Bad Sector Retest operation lets you check sectors on FAT partitions that have been marked bad and recover sectors that are usable.

The FAT file system allocates disk space for file storage in units called clusters, which are composed of a fixed number of sectors. Because the FAT file system tracks bad sectors at the cluster level, it marks an entire cluster bad even though the problem may exist in a single sector.

Use Info to discover whether a partition contains bad clusters. For more information, see “Getting Information About Partitions” on page 98.

As a conservative measure, when you move or resize a partition or increase cluster size, PQDisk marks all new clusters containing any part of old bad clusters as bad (even though the clusters may not actually contain bad sectors). Likewise, when you decrease a partition's cluster size, PQDisk divides bad clusters into multiple bad clusters. If, after you complete these tasks, PQDisk reports bad sectors, you can perform Bad Sector Retest and reclaim the good sectors that were marked bad.

WARNING! Some sectors marked as bad are “marginally bad,” meaning that one time the sector works fine and another time it does not. Bad Sector Retest may mark a marginally bad sector as good. This can result in data loss if the marginally bad sector fails in the future. Most modern hard drives detect bad sectors and automatically remap the sector, so in general, you do not see bad sectors on modern hard drives. If you do get bad sector errors on a modern hard drive, it is recommended that you replace the drive.

Hiding and Unhiding Partitions

The Hide Partition operation lets you secure partitions against unwanted user access. You can perform this operation on FAT, FAT32, and NTFS partitions. When you hide a partition, the next time you boot your computer the partition is not assigned a drive letter. Conversely, when you unhide a partition, the next time you boot your computer the partition is assigned a drive letter.

WARNING! Unless you are running Windows NT/2000/XP, un-hiding multiple primary partitions may cause data loss. Hiding and unhiding partitions can cause the drive letters of other partitions to change. When this happens, your computer may not boot and applications may not run.

If your hard disk has more than one primary partition, only one is visible by default. When you use the Set Active command, PQDisk unhides the selected primary partition. Unlike Norton PartitionMagic, PQDisk will not hide the other primary partitions.

If you are running Windows NT/2000/XP, you can have multiple visible primary partitions without causing data loss problems.

Resizing the Root Directory

The Resize Root operation lets you change the maximum number of entries that can be placed in the root directory of a FAT partition. The number of root entries is set at the time the partition is formatted; the limit does not expand automatically as it does in a subdirectory or in a FAT32 partition. Consider increasing this number if you use Microsoft long filenames in the root directory. During this operation, data within the partition is unaffected.

Occasionally, enlarging the root directory displaces the first few files on the partition (such as IO.SYS and MSDOS.SYS if the partition contains an operating system). If the root directory is on a boot partition and the partition fails to boot after resizing the root directory, you can run SYS.COM to move the displaced files back to the front of the disk.

Setting an Active Partition

The Set Active operation lets you make a partition the active partition (the partition the computer boots from). Only one partition can be active at a time. To boot your computer from a partition, the partition must be on the first disk, and it must contain an operating system. When your computer boots, it reads the partition table of the first disk to find out which partition is active and boots from that partition.

IMPORTANT! Before you make a partition active, it must be bootable. If the partition is not bootable or if you are not certain if it is, have a boot diskette ready.

Unlike Norton PartitionMagic, PQDisk does not hide inactive FAT and NTFS primary partitions. Windows 95, Windows 98, and DOS FDISK programs cannot hide or unhide partitions. Hiding inactive primary partitions makes it easy to install multiple operating systems and choose the one you want to use with Set Active. For example, if you have Windows 98 and want to install Windows NT, 2000, or XP in a separate partition, you can make the Windows 98 partition smaller, create another primary partition, set it as the active partition, and then boot from the Windows NT/2000/XP installation diskettes.

Windows NT, 2000, XP. In a configuration with mixed IDE and SCSI hard disks, Windows NT/2000/XP does not always see the boot drive as the first disk.

Resizing Clusters

The Resize Clusters operation lets you change the cluster size on FAT and FAT32 partitions. Reducing cluster size may help you reclaim wasted space on your hard disk. All files on FAT and FAT32 partitions are stored in allocation units called clusters. Each file on a partition is allotted at least one cluster. The size of a partition determines cluster size. Unless the size of a file is an exact multiple of cluster size on the partition where the file is located, the file includes wasted space. Larger partitions have larger clusters, and, therefore, more wasted space.

PQDisk adheres to the established limits for partition and cluster sizes. You cannot select a cluster size that is invalid for the selected partition. Choosing a smaller cluster size may resize the partition smaller, creating unallocated space next to the partition. You can use this unallocated space by creating a new partition.

IMPORTANT! Do not choose the 64 K cluster size unless you have Windows NT/2000/XP and a 2-4 GB disk.

Default Cluster Sizes

A partition's cluster size is set by the DOS FORMAT operation, based on the size of the partition, as shown in the following tables.

It is not recommended that you use the smallest cluster size on partitions containing a single, large file, such as a database or swap file.

DOS and Windows default FAT cluster sizes

Partition Size (MB)	FAT Type	Sectors Per Cluster	Cluster Size
0-15	12-bit	8	512 bytes
16-127	16-bit	4	2 K
128-255	16-bit	8	4 K
256-511	16-bit	16	8 K
512-1,023	16-bit	32	16 K
1,024-2,047	16-bit	64	32 K
2,048-4,096	16-bit	128	64 K*

*Only available with Windows NT and Windows 2000 and a 2–4 GB disk.

Windows 95 OSR2, Windows 98, Windows Me, and Windows 2000 default FAT32 cluster sizes

Partition Size (GB)	Sectors Per Cluster	Cluster Size
0.256- 8.01	8	4 K
8.02-16.02	16	8 K
16.03-32.04	32	16 K
> 32.04	64	32 K

Converting Partitions

The following partition conversions are supported by PQDisk.

- Converting FAT Partitions to FAT32
- Converting FAT32 Partitions to FAT
- Converting NTFS Partitions to FAT
- Converting NTFS Partitions to FAT32

You cannot use PQDisk to convert FAT partitions to NTFS.

Converting FAT Partitions to FAT32

The Convert FAT to FAT32 operation converts a FAT partition to FAT32. FAT32 partitions have less wasted disk space than FAT partitions. However, you should be aware of these issues:

- You must have Windows 95 OEM Service Release 2, Windows 98, Windows Me, Windows 2000, or Windows XP to access files on a FAT32 partition. If you run an operating system other than these, FAT32 partitions will be inaccessible when the other operating system is running, even if one of these operating systems is installed on your machine.
- Some laptops have a sleep mode that saves all memory to disk. Because this function sometimes requires a FAT partition, consult your laptop manual or contact the manufacturer before converting to FAT32.
- The minimum recommended size for a FAT32 partition is 256 MB.
- You should not convert compressed drives. First uncompress the drive and then convert the file system type. Then compress the drive again if necessary.

Converting FAT32 Partitions to FAT

The Convert FAT32 to FAT operation converts a FAT32 partition to FAT.

To complete this conversion, the partition must have at least 300-400 MB of unused space because of how the FAT file system allocates disk space for file storage.

The command will generate an error and write it to the log file if your FAT32 partition contains over 2 GB of data. If the partition size is over 2 GB but it contains less than 2 GB of data, you can convert the partition (without data loss), but the new partition will be 2039 MB. At this point, PQDisk may report too many root directory entries (the maximum number of entries in a FAT partition's root directory is limited, unlike a FAT32 partition's root directory). In this case, move or copy some of the files in the root directory to another location and then start the conversion again.

Converting NTFS Partitions to FAT or FAT32

The Convert NTFS to FAT operation converts an NTFS partition to FAT. The Convert NTFS to FAT32 operation converts an NTFS partition to FAT32.

Converting an NTFS partition to FAT allows you to view the contents of the partition from DOS, Windows 95, Windows 98 or Windows Me. Converting an NTFS partition to FAT32 allows you to view the contents of the partition from Windows 95B, Windows 98, or Windows Me.

IMPORTANT! You will lose file system-specific information when converting from NTFS to FAT or FAT32. Refer to “NTFS Information Lost When Converting to FAT or FAT32” on page 105 for additional information.

NTFS is an advanced version of FAT and FAT32. Therefore, depending on the NTFS features used on the partition, the type of data, and partition size, you may or may not be allowed to complete the conversion.

If you receive an error message and the conversion stops, it is usually caused by one or more of the following:

- The file system for conversion is not allowed for the current partition size. A FAT32 partition should be greater than 256 MB, and a FAT partition must be less than 2 GB.
- The NTFS partition has data in memory that has not yet been written to the hard disk.
- The file system has errors, such as lost clusters and cross-linked files. You can fix these problems, then try the conversion again.
- There is not enough temporary space in the partition to do the conversion. The conversion requires both the NTFS system and the FAT32 system files be present until the last step of the conversion. Also, there is data in NTFS FRS's that must be moved to external clusters and saved.

NTFS Information Lost When Converting to FAT or FAT32 If you can complete the conversion from NTFS to FAT or FAT32, you may receive a warning about the quality of data and feature loss, depending on the features used on the partition, the type of data, and the partition size.

Warning	Description
Error	The conversion is not allowed. Because the partition being converted is using advanced features in NTFS, you may experience unintended data and feature loss. You will receive an error in one or more of the following cases: • There is more than one data stream for any file. • Any links. • Any extended attributes. • Any user-defined attributes in any file. • Device entries. • There are sparse files on the volume. Any sparse files, except for the bad sector file, will stop the conversion.

Warning	Description
Warning	The conversion is allowed. Although a conversion warning is not as serious as an error, you may still experience the loss of NTFS-specific features that are not supported in FAT or FAT32. You will receive a conversion warning in one or more of the following cases: • Disk usage quotas - NTFS supports limiting the amount of disk space for a user. After conversion, all users will have full access to all free hard disk space. • Access control lists - This is a file attribute that lists all the users that can access a file. After conversion, all users will have full access to all files. • Index of access control lists - A list of all files that have specific access rights assigned to them. After conversion, all users will have full access to all files. • FAST index file - This file is sometimes created on Windows 2000 computers. After conversion, all indexing of keywords will be lost. • Old versions of files - NTFS has the ability to keep versions of files, however, only the current version of the file is converted and saved.
No Warning	The conversion is allowed. The most basic NTFS partition still gives files more features than are found in FAT or FAT32. When Windows NT 4.0 is used to copy files from an NTFS partition to a FAT partition, no warning is given about the features you are losing. Also, the conversion will not give you a warning about specific features that cannot be converted. These features include: • Standard journal file (only used internally by NTFS) - This file is a transaction log of changes to the NTFS file system. After conversion, the journal file will be lost. • NTFS-specific file attributes - NTFS and FAT both have standard file attributes, such as Read-only, Archive, Hidden, and System. NTFS has additional file attributes that can be set. After conversion, however, these additional file attributes will be lost. • NTFS-specific file dates - The last edit date is converted to the FAT date. After conversion, the creation date, last access date, and last edit date (date change only) will be lost. • Reliable change journal - This journal file is new to Windows 2000. After conversion, this file will be lost.

Command Line Switches

To run PQDISK specify PQDISK /switch, where switch is one or more of the available command line switches, you must always specify the /CMD switch. All other switches are optional and are not order-sensitive.

Switch **Description**

/CMD= Designates the name of the script text file. For example, if the script file were named SCRIPT.TXT, the command line syntax with the script would be:

```
PQDISK /CMD=SCRIPT.TXT
```

/DBG Outputs debugging information to PQ_DEBUG.TXT.

/DSS *(Display Script Status)* Causes all output to be written to the screen. This switch also allows progress indicators to be displayed. Use the /DSS switch as follows:

```
PQDISK /CMD=SCRIPT.TXT /DSS
```

/LOG= Designates the name of the log file where all output is directed. The log file records information about each script command as the command is executed. To generate a log file named USERLOG.TXT the following command line syntax would be used:

```
PQDISK /CMD=SCRIPT.TXT /LOG=USERLOG.TXT
```

WARNING! Do not attempt to store the log file on any partition that will be modified. If you do, you will damage your partition.

/ERR= Designates the name of the file where error information is written. When an error occurs the error number is written to the error file. Because the error file contains only the error number, without its associated text description, it can be easily parsed by an external, user-defined application. If the /ERR switch is used and the program terminates without an error the error file is not created. The following syntax illustrates how to specify an error file named ERRORLOG.TXT:

```
PQDISK /CMD=SCRIPT.TXT /LOG=USERLOG.TXT /ERR=ERRORLOG.TXT
```

/F4K Forces 4 K clusters on FAT32 partitions that are larger than 8 GB. Converting FAT32 partitions to NTFS is much faster when the partition is aligned with 4 K clusters.

/IHF Skips the initial check for hibernation files, reducing the time PQDisk needs to initialize.

Switch	Description
/INFO	Displays information about all the drives in a computer.
/NRF=	Specifies the name of a file, which if it exists, will prevent the script from executing. You can use this switch in a login script or autoexec.bat to prevent PQDisk from running a second time. For example, you could use the following command line to specify a script and a log file and prevent PQDisk from running if the log file already existed. `PQDISK /CMD=SCRIPT.TXT /LOG=USERLOG.TXT /NRF=USERLOG.TXT` The /NRF parameter can be specified more than once on the command line when it is necessary to check for more than one file.
/POC	Checks the order of the primary partitions on each disk against the order of partitions in the MBR. When you use this switch, PQDisk displays the volume, file system, size, status, and start sector for each primary partition on the drive. The partitions are displayed in MBR order. A message is displayed if any of the start sectors do not match the order in the MBR. A separate message is displayed for each drive where the order is not correct. If there is a problem with the order, you can fix it with the /POF switch. If the order is correct, no message will display.
/POF	Changes the order of primary partitions in the MBR to match the order of the partitions on disk. When you use this switch, PQDisk displays the volume, file system, size, status, and start sector for each primary partition on the drive. The partitions are displayed in MBR order. If any of the start sectors did not match the order in the MBR, a message is displayed saying that the order did not match but has been corrected. A separate message is displayed for each drive, where applicable. If the order is correct, no message will display.
/SCO	Specifies to only check the syntax of a script. It will verify that a partition is selected before an operation is executed and check the syntax of all the script commands. It will also ensure that any volume labels specified in a select partition command are unique. It will not actually run the script. The syntax check will not detect logical errors such as trying to move the partition when there is not free space. This parameter can be used with the /LOG file if desired. A successful syntax check will show a command saying that it was successful. Use the /SCO switch as follows: `PQDISK /CMD=SCRIPT.TXT /SCO`
/VER	Displays the version of PQDisk.

Scripting Overview and Syntax

Each script command must be contained within a single line and may not wrap to subsequent lines. Script lines may include valid script commands, blank lines, and commented text. The PQDisk scripting language uses C++ style comments that begin with the "//" characters. A single script line is limited to 180 characters.

It is recommend that each of the partitions to be modified be checked at the first of the script. Because the script file will terminate as soon as an error occurs, checking each of the partitions first will keep the script from actually making any changes when errors are found.

Because partitions must start and end on cylinder boundaries, when a number like 10 (MB) is specified, the value actually used will be rounded up to the size that fits to the end of the cylinder boundary. This size difference depends on the geometry of the drive.

When specifying a partition size, a margin of error of 1 cylinder above or below that amount is allowed (a range of 2 cylinders centered on the amount specified.) This means if an amount like 10 (MB) was specified for a resize, and a cylinder was .5 (MB), that the operation would complete successfully if it can resize the partition to at least 9.5 (MB). The actual range would be 9.5 to 10.5 (MB). If it could not resize the partition within this range, it will return an error.

Use caution when selecting a partition by number. This feature is available to select free space or partitions that have no drive letter or label. The problem with selecting a partition by number is that the numbers can and will change throughout the script execution. If I select partition #2 and move it to the right, any free space that has been moved from the right to the left side of the partition will now become partition #2. (The partition moved will still be selected regardless of the number).

Using the Select Partition Next and Previous commands are usually preferable to selecting free space by partition number. With most operations, the partition selected after an operation will be the partition operated on. For example, Resize and Move will always leave the partition operated on as the selected partition after the operation. On a Create command, the partition created will be selected after the command whether it is at the beginning or end of the free space it was created in.

Script Commands

Several special characters are used when describing the syntax of script file commands:

Character	Denotes
{}	A required parameter
[]	An optional parameter
|	A choice among two or more options

If the syntax for a command includes several parameters, the command description includes a syntax example. If there is no syntax example, the command does not include parameters. For example, Bad Sector Retest does not include parameters, but Cluster Analyzer does.

Bad Sector Retest

Retest the current partition for bad sectors and unmark any bad sectors that have been set incorrectly.

Check

Use to check a selected partition for errors.

No parameters.

Cluster Analyzer

Syntax: `Cluster Analyzer [/ClusterSize={512|1|2|4|8|16|32|64}] [/ShowClusterWaste][/SetToRecommended]`

Get Cluster Analysis information about a particular partition.

If this operation is used with out any parameters, it will output a cluster analysis screen.

Parameter	Description
/ClusterSize	Changes the cluster size to the specified size.
/ShowClusterWaste	Shows the Cluster Analysis Screen.
/SetToRecommended	Sets the ClusterSize to the recommended size.

Convert To

Syntax: `Convert to {FAT|FAT32}`

The Convert command changes a FAT partition to FAT32, a FAT32 partition to FAT, and an NTFS partition to FAT or FAT32. No parameters.

Copy

The Copy command makes a copy of the selected source partition to the selected destination or copy partition. It must be preceded by the following commands:

- Select Drive {Num}
- Select Partition {PartitionLetter|“Volume Label”|Extended|Next|Previous|Num }
- Select Copy Drive {Num}
- Select Copy Partition {PartitionLetter|“Volume Label”|Next|Previous|Num }

Select the partition to copy using the Select Drive and Select Partition commands. Select the destination free space using the Select Copy Drive and Select Copy Partition commands. If the selected partition is a valid partition and the partition specified by the copy partition is a block of free space large enough to hold the partition, the copy operation will copy the selected partition to the specified free space.

The copy will be the same size as the source partition. You can use the Resize command to change the size of the new partition.

Create

Syntax: `Create /FS={FAT|FAT32|LINUXEXT2|LINUXEXT3|LINUXSWAP|NTFS|EXTENDED| UNFORMATTED} [/Label=\"NEW LABEL"] [/Size=Value] [/Position={BEGINNING|END}]`

Create a new partition, and, optionally, format it.

If you create a 7.8 MB partition below the 1024 cylinder boundary, it will be marked as FAT12; if you create a 7.8 MB partition above the 1024 cylinder boundary, it will be marked FAT16X. This behavior matches the way NT DiskAdministrator marks partitions.

Parameter	Description
/FS	*(Required)* It can be any of the above specified strings. There may be cases where creating with a certain /FS would fail. For example, if you tried to create an extended partition when one already existed, the operation would fail.
/Label	*(Optional)* Replace "NEW LABEL" with the desired volume label. It must be 11 characters or less for FAT partitions. Labels must be 16 characters or less for Linux Ext2 and Ext3 partitions and 32 characters or less for NTFS partitions. The label must be in double quotes. The script may fail if invalid characters are entered.
/Size	*(Optional)* Specified in megabytes and will default to the size of the free space if not specified. FAT32 partitions cannot be larger than 192 MB.
/Position	*(Optional)* Must be followed by either END or BEGINNING to specify where the partition will be created in the free space.

All primary partitions that you create will be visible unless you explicitly hide them.

Delete

Syntax: `Delete [All|"volume label"]`

Deletes a partition. If DELETE ALL is specified, all of the partitions on the currently selected disk are deleted. If no parameters are specified, it will delete the selected partition on the selected drive. If no partition is selected, an error will be written to the log file.

Parameter	Description
ALL	Deletes all partitions on the selected drive. If you used the Protect Partition command to protect a partition, it is not deleted by Delete All. See page 116.
"volume label"	If you specify a volume label, the partition will not be deleted unless the volume label matches this command. If no volume label is specified, the selected partition will be deleted without confirming the volume label.

Format

Syntax: `Format ["volume label"] /FS={ FAT|FAT32|LINUXEXT2|LINUXEXT3|LINUXSWAP| NTFS } [/Label="NEW LABEL"]`

Parameter	Description
"volume label"	If you specify a volume label, the partition will not be formatted unless the volume label matches this command. If no volume label is specified, the selected partition will be formatted without confirming the volume label.
/FS	*(Required)* Can be any of the above specified strings. There may be cases where formatting with a certain /FS would fail, for example trying to format a FAT partition past 1024 cylinders.
/Label	*(Optional)* Replace "NEW LABEL" with the desired volume label. It must be 11 characters or less for FAT partitions. Labels must be 16 characters or less for Linux Ext2 and Ext3 partitions and 32 characters or less for NTFS partitions. The label must be in double quotes. The script may fail if invalid characters are entered.

Hide

Hide the currently selected partition. This operation changes the partition type value by ORing in a value of 0x10. This makes the partition type unknown for Windows 9x and Window Me and no drive letter will be assigned. However; Windows 2000 and Windows XP will be able to see the partition. Using this command on a partition only affects that partition. If you hide the active partition, there will not be an active partition on the disk.

See also, "Unhide."

Info

Syntax: `Info [/Usage][/Waste][/Partition][/FS]`

Display information about a selected partition. If you do not specify any parameters, the Info command displays all the information about the selected partition.

Parameter	Description
/Usage	The Disk Usage screen is available for the FAT, FAT32, and NTFS file systems. This screen shows you the following information in bytes, megabytes, and as a percentage: • Used space on the partition, including space wasted by clusters • Free space on the partition • Bad space on the partition • Total space on the partition (found by adding the three previous lines)

Parameter	Description
/Waste	The Cluster Waste screen applies only to partitions that use either the FAT or FAT32 file system. This screen shows the following: • Current cluster size in bytes or kilobytes • Data stored on the partition in bytes and megabytes • Wasted space on the partition in bytes and megabytes Total Used space in bytes and megabytes (found by adding the numbers on the two previous lines)
/Partition	This screen is available for all types of partitions, including free space and extended partitions. Information on this tabbed page includes the following: • Partition type is shown in hexadecimal followed by a text description of the partition or file system type (such as FAT, FAT32, NTFS, and so on). The hexadecimal designation is a conventional way to display partition types. • Serial number is shown here if the partition's file system uses serial numbers. Not all file systems use serial numbers. The next section of the screen shows physical information about the partition, including the following: • First Physical Sector shows the logical number and the location (cylinder, head, and sector) where the partition begins. • Last Physical Sector shows the logical number and the location (cylinder, head, and sector) where the partition ends. • Total Physical Sectors gives the number of sectors in the partition. • Physical Geometry shows the number of cylinders, heads, and sectors of the physical disk drive on which the partition resides.

Parameter	Description
/FS FAT	This screen applies only to partitions that use the FAT or FAT32 file system. The first section on this page provides the following information about the file system: • Sectors per FAT • Root directory capacity • First FAT sector • First data sector The next section of this page gives the following information: • Number of bytes in files on the partition, the number of files, and the number of those files that are hidden • Number of bytes in directories on the partition, the number of directories, and the number of those directories that are hidden Several extensions to the FAT file system exist. The final section of this page gives the following information about FAT extensions: • Number of bytes used for OS/2 Extended Attributes and how many files and directories the Extended Attributes are associated with Number of bytes used for long filenames and the number of files and directories the long filenames are associated with
/FS NTFS	This screen contains information that pertains only to NTFS partitions. This screen shows the following file system information for the selected partition: • NTFS Version shows the version number. The most recent version is 1.2. • Bytes per NTFS sector displays the number of bytes in each logical sector on the selected partition. (There are always 512 bytes in each physical sector.) • Cluster size • First MFT Cluster • File Record Size The next section shows information similar to that shown by NTFS CHKDSK, including the following: • Number of files and the bytes and clusters allocated to them • Of the clusters used in files, the number of wasted bytes resulting from the cluster size • Number in indexes (directories) and the space allocated to them, shown in bytes and clusters • Space reserved for other system structures, shown in both bytes and clusters

Label

Syntax: `Label [/GetLabel ] [/SetLabel=""]`

This option lets you change the name of a selected partition. Labels can be up to 11 characters long for FAT, and FAT32 partitions, 32 characters long for NTFS partitions, and 16 characters long for Linux Ext2 or Exte3 partitions. Labels for FAT and FAT32 partition types follow the same rules as DOS names, with two exceptions: spaces are allowed, and no period is required between the first eight characters and the last three. The label must be enclosed in double quotes.

Move Left

Syntax: `Move Left {Max|Min|Value}`

Move a partition to the left. If the partition is the extended partition, only the right boundary is changed.

Parameter	Description
Max	Move the partition as far to the left as possible. Flush with the previous partition or beginning of the drive.
Min	Move the partition to the left, the minimum amount possible (1 cylinder).
Value	Move the partition left by the amount of the value specified (in megabytes).

Move Right

Syntax: `Move Right {Max|Min|Value }`

Move a partition to the right. If the partition is the extended partition, only the right boundary is changed.

Parameter	Description
Max	Move the partition as far to the right as possible. Flush with the previous partition or beginning of the drive.
Min	Move the partition to the right the minimum amount possible (1 cylinder).
Value	Move the partition right by the amount of the value specified (in megabytes).

Move Space Before

Syntax: `Move Space Before {Max|Value }`

Same as Move Right.

Parameter	Description
Max	Same as Move Right Max. Makes as much space before the partition as possible by moving the partition right.
Value	Moves the partition right such that the space before is equal to the value specified if possible (specified in megabytes).

Move Space After

Syntax: `Move Space After {Max|Value}`

Same as Move Left.

Parameter	Description
Max	Same as Move Left Max. Makes as much space before the partition as possible by moving the partition right.
Value	Moves the partition left such that the space before is equal to the value specified if possible (specified in megabytes).

Protect Partition

Syntax: `Protect Partition {Letter|"Label"|Number}`

Protects the specified partition from being deleted by the Delete All command.

Parameter	Description
Letter	Drive letter of the desired partition.
"Label"	Volume label of the desired partition. It must be in quotes. PQDisk searches the currently selected drive for a volume with a matching label.
Number	Number of partition or unallocated space. This command should always be preceded by a Select Drive command. Always select unallocated space by its partition number. The first partition or block of unallocated space on each drive is number 1. You can display partition numbers with the Show Partitions command.

Reboot

Reboot the computer immediately. This command may be used without first selecting a drive or partition.

Resize

Syntax: `Resize  {Max|Min|Value } [/Clustersize={512|1|2|4|8|16|32|64}]`

Resize a partition.

Parameter	Description
Max	Resizes to the maximum size possible. The right edge of the partition will be flush with end of drive or next partition, if possible.
Min	Resizes to the minimum possible (determined by the size of the data).
Value	Resize to value specified (in megabytes).
/Clustersize	Will set the cluster size to the size specified during the resize operation. The cluster size must be valid for the partition size specified.

Resize Larger

Syntax: `Resize Larger {Max|Min|Value}  [/Clustersize={512|1|2|4|8|16|32|64}]`

Resize a partition larger by specifying the incremental change in size.

Parameter	Description
Max	Same as Resize Max. Make the partition as big as possible.
Min	Grows the size of the partition by the smallest amount possible (1 cylinder).
Value	Grows the size of the partition by the size specified (in megabytes).
/Clustersize	Sets the cluster size to the size specified during the resize operation. (The cluster size must be valid for the partition size specified.)

Resize Left Boundary

Syntax: `Resize Left Boundary {Max|Min|Value }`

Resize the extended partition by moving the left boundary. This operation is for extended partitions only. .

Parameter	Description
Max	Resizes to the maximum size possible. The right edge of the partition will be flush with end of drive or next partition, if possible.
Min	Resizes to the minimum possible (determined by the size of the data).
Value	Resize to value specified (in megabytes).

Resize Left Boundary Larger

Syntax: `Resize Left Boundary Larger {Max|Min|Value }`

Resize an extended partition larger by specifying the change in position of the left boundary. This operation is for extended partitions only.

Parameter	Description
Max	Same as Resize Left Boundary Max. Make the partition as large as possible.
Min	Grow the size of the partition by the smallest amount possible (1 cylinder).
Value	Grow the size of the partition by the size specified (in megabytes).

Resize Left Boundary Smaller

Syntax: `Resize Left Boundary Smaller {Max|Min|Value }`

Resize an extended partition larger by specifying the change in position of the left boundary. This operation is for extended partitions only. See also, "Resize Left Boundary."

Parameter	Description
Max	Same as Resize Left Boundary Min. Make the partition as small as possible.
Min	Partition's size will be decreased by the minimum amount possible (1 cylinder).
Value	Partition's size will be decreased by the amount specified (in megabytes).

Resize Smaller

Syntax: `Resize Smaller {Max|Min|Value}   [/Clustersize={512|1|2|4|8|16|32|64 }]`

Resize a partition smaller by specifying the incremental change in size. See also “Resize” on page 116.

Parameter	Description
Max	Same as Resize Min. The partition will be as small as possible.
Min	Partition's size will be decreased by the minimum amount possible (1 cylinder).
Value	Partition's size will be decrease by the amount specified (in megabytes).
/Clustersize	Sets the cluster size to the size specified during the resize operation. The cluster size must be valid for the partition size specified.

Resize Space After

Syntax: `Resize Space After {Max|Min|Value}`
`[/Clustersize={512|1|2|4|8|16|32|64}]`

Resize a partition by specifying the free space desired after the partition after the resize is completed. See also “Resize” on page 116.

Parameter	Description
Max	Resizes so that the space after the partition is as large as possible. The partition is as small as possible.
Min	Resizes so that the space after the partition is as small as possible. The partition is as large as possible.
Value	Sizes the partition such that the space after is the size of value (in megabytes).
/Clustersize	Sets the cluster size to the size specified during the resize operation. Cluster size must be valid for the partition size specified.

Resize Space Before

Syntax: `Resize Space Before {Max|Min|Value }`

Resize an extended partition by specifying the free space desired before the partition after the resize is completed. This operation is for extended partitions only.

Parameter	Description
Max	Resizes so that the space before the partition is as large as possible. The partition is as small as possible.
Min	Resizes so that the space before the partition is as small as possible. The partition is as large as possible.
Value	Sizes the partition such that the space before is the size of value (in megabytes).

Resize Clusters

Syntax: `Resize Clusters {Value|Min|Max }`

Change the cluster size of a FAT partition.

Parameter	Description
Value	Can be either 512,1,2,4,8,16,32 or 64. If the cluster size specified requires the partition to change size, the Resize Clusters operation will resize the partition as well.
Min	If specified, Min will change the partition to the minimum partition size possible. This may also require the partition size to change. If needed, the partition size will be changed.
Max	If specified, the partition will be modified to use the largest cluster size possible.

Resize Root

Syntax: `Resize Root {Value|Min|Max}`

Change the number of entries in the root directory of a FAT partition.

Parameter	Description
Value	Must be a value between 512 - 1024. This will change the maximum number of root entries possible for this partition. The number actually set will be the closest number possible to the number specified.
Min	Sets the partition to having the smallest possible maximum number of root entries.
Max	Sets the maximum number of root entries for the partition to the largest number possible.

Select Copy Drive

Syntax: `Select Copy Drive {Num}`

Select the copy destination drive from among the physical drives.

Parameter	Description
Num	Number of the drive to select starting at 1.

Select Copy Partition

Syntax: `Select Copy Partition {PartitionLetter|"Volume Label"|Next|Previous|Number}`

Select the copy destination free space block.

Parameter	Description
PartitionLetter	Drive letter of the desired partition.
"Volume Label"	Volume label of the desired partition. It must be in quotes. PQDisk searches the currently selected copy drive for a volume with a matching label.
Next	Must have had a partition selected previously.
Previous	Must have had a partition selected previously.
Number	Number of the free space block. This command should always be preceded by a Select Copy Drive command. Always select the free space by its partition number. The first partition or free space block on each drive is number 1. Partition numbers are displayed by the Show Partitions command.

Use caution when selecting a partition by number, because the numbers change as operation are performed. Once selected by number, it remains selected even if a subsequent operation changes the number of the selected partition.

Select Drive

Syntax: `Select Drive {Num}`

Select the drive to act upon from among the physical drives. This command fails if there are partition table errors on the selected drive.

Parameter	Description
Num	Number of the drive to select starting at 1.

Select Partition

Syntax: `Select Partition {PartitionLetter|"Volume Label"|Extended|Next|Previous |Number}`

Select the partition or free space block to act upon.

Parameter	Description
PartitionLetter	Drive letter of the desired partition.
"Volume Label"	Volume label of the desired partition. It must be in quotes. PQDisk searches the currently selected drive for a volume with a matching label.
Extended	Specifies the extended partition of the selected drive. You must use a Select Drive command prior to this command.
Next	Must have had a partition selected previously.
Previous	Must have had a partition selected previously.
Number	Number of the free space block. This command should always be preceded by a Select Drive command. Always select the free space by its partition number. The first partition or free space block on each drive is number 1. Partition numbers are displayed by the Show Partitions command. This is the only way to select free space for creating a new partition.

Use caution when selecting a partition by number, because the numbers change as operation are performed. Once selected by number, it remains selected even if a subsequent operation changes the number of the selected partition.

Set Active

Mark the selected partition as the active, or bootable, partition. Other primary partitions are not hidden.

Set Default Bad Sector Test State { ON|OFF}

Syntax: `Set Default Bad Sector Test State {ON|OFF}`

Set the bad sector testing ON or OFF for all partitions on the currently selected drive. If set on, then all sectors to be used in a partition creation or resize operation will be tested before the creation or resize operation will be performed.

Bad sector testing can take large amounts of time for large partitions. With old drives this will ensure the correct operation of the disk drive sectors used. With new hard drives this test rarely provides any benefit and should be set to off.

Set Drive Read Only Mode

Syntax: `Set Drive Read Only Mode {ON|OFF}`

Set the read-only flag ON or OFF for all partitions on the currently selected drive. When set on for a drive, modifications to the partitions on that drive will not be allowed.

Set Ignore OS/2 EA Errors

Syntax: `Set Ignore OS/2 EA Errors {ON |OFF}`

This option tells whether or not to ignore OS/2 extended attribute errors when it checks a FAT partition.

Set NT 64K FAT Clusters

Syntax: `Set NT 64K FAT Clusters { ON|OFF }`

This option enables or disables handling of FAT partitions using 64K cluster sizes. Because only Windows NT/2000/XP can handle FAT partitions with 64K cluster sizes, this ability can be disabled.

Set Partition Table Autofix

Syntax: `Set Partition Table Autofix { ON|OFF }`

This option enables or disables the option for PQDisk to check for partition table errors when the Select Drive or Select Copy Drive command is used. If this command is used and set to ON PQDisk will fix any partition table errors it can. By default, autochecking and autofixing are disabled.

If you copy a partition to a drive with partition table errors, the operation will fail and PQDisk may crash with a DOS4GW or a causeway exception.

Show Copy Partitions

The Show Copy Partitions command should be preceded by the following commands:

Select Drive {Number}
Select Partition {PartitionLetter|"Volume Label"|Extended|Next|Previous|Number}
Select Copy Drive {Number}

For the Show Copy Partitions command to work correctly, a drive and partition need to be selected and a copy drive needs to be selected.

Show Partitions

Displays partition information for all partitions on the selected drive. If the /LOG or /DSS switches are used, the following tables will be written to the log file or the screen:

```
Volume          Type         Size MB  Used MB  Free MB   Status     Pri/Log
--------------- ------------ -------- -------- --------  ---------- -------
1 - *:WINNT4    Hid.FAT16B      902.1    399.1    503.0  Hidden     Primary
2 - *:WIN95C    FAT32           706.0    124.5    581.5  Active     Primary
3 - *:WINXP     Hid.FAT16B    2,039.5    622.0  1,417.5  Hidden     Primary
4 - *:          Hidden FAT      549.1      7.2    541.9  Hidden     Primary
5 - *:          Free Space   15,366.9      0.0 15,366.9  None       Primary
```

```
Volume            Type         Size MB  Used MB  Free MB   Status     Pri/Log
---------------   ------------ -------- -------- --------  ---------- -------
1 - *:SWAPSPACE2  Linux Swap      141.2      0.0    141.2  None       Primary
2 - *:/           Linux Ext2    1,717.9    952.6    765.3  None       Primary
3 - *:            Extended      4,290.8  4,290.8      0.0  None       Primary
4 - *:TOOLS       FAT16B        1,921.8  1,552.8    369.0  None       Logical
5 - *:DATA        Hidden FAT32  2,368.9  1,698.2    670.8  Hidden     Logical
```

Show Preferences

This command displays the current preference settings. These settings include FAT 64K cluster handling, bad sector testing, and the drive read-only state.

Unhide

Unhide the currently selected partition. This clears the 0x10 bit in the partition type. If you use this command in conjunction with the /DSS switch, PQDisk will unhide multiple visible primary partitions without issuing a warning or displaying a prompt.

See “Hide” on page 112.

Script File Examples

Several script file examples are given in this section to demonstrate how to use the PQDisk script language.

- Resize and Move Partitions
- Create Partitions
- Resize and Create Partitions
- Cluster Analyzer
- Copy a Partition
- Partition Information
- Delete Partitions
- Delete All Partitions
- Format Partition
- Hide, Unhide, and Set Active
- Show Partitions

Scenario 1 - Resize and Move Partitions

The first physical drive is partitioned into three visible primary partitions, C:, D:, and E:. There is no unallocated space on the disk. We want to take 1000MB from D: and add it to E:

```
// Check the partitions to be operated on first
Select Drive 1
Select Partition D
Check
Select Partition E
Check

// Select the first partition I want to change
Select Partition D

// Shrink the partition by 1000 megabytes
Resize Smaller 1000

// Select the partition to add the 1000 megabytes to
Select Partition E

// Move the partition as far as possible to the left so that

// the free space just created will be on the left edge (the end)
Move Left Max

// Take up all of the available space
Resize Larger Max
```

Scenario 2 - Create Partitions

The first physical drive is 40 GB in size and has one large NTFS C: partition.   The C: partition contains only 2300 MB of data so we would like to take some space from C:, create a 20 GB logical E: partition and give the remaining space to a logical E: partition.

```
// Check the partition first
```

```
Select Drive 1
Select Partition C
Check

// Partition C is already selected so shrink it to 5000 MB
Resize 5000

// Since the C partition is still selected after the resize, we need
to select
// the free space after C.
Select Partition Next

// Create the extended partition to the default size, which will be
// all of the free space currently selected.
Create /FS=EXTENDED

// The Extended partition is now selected, and we want to select the
// next free space in the extended partition.
Select Partition Next

// Create the partition that we need to be 20000 MB first at the end
// of the free space that is currently selected. (What will be the
// E partition)
Create /FS=FAT32 /Label="DBFILES " /Size=20000 /Position=END

// Select the rest of the free space within the extended partition.
// Since the last partition was created at the end of the free space,
// we need to move to the free space previous to the selected
// partition.
Select Partition Previous

// Create the partition in the rest of the free space
Create /FS=FAT32 /Label=" DATA"
```

Scenario 3 - Resize and Create Partitions

The first physical drive contains a primary C: partition and two logical partitions (D: and E:) within an extended partition. There is no unallocated space on the disk. We want to create a logical F: partition by taking 4000 MB of space from the C: partition. The F: partition will be a FAT32 partition with a volume label of "DATA." Because the drive is fairly new, we skip bad sector testing for all operations.

```
// Check all of the partitions first
Select Drive 1
Select Partition C
Check
Select Partition D
Check

Select Partition E
Check

// Since a partition on this drive had already been selected, we
// can set the default bad sector testing to off for this drive
Set Default Bad Sector Test State Off
```

```
// Select the C: partition and shrink it by 4000 MB
Select Partition C
Resize Smaller 4000

// Select the extended partition and resize the left boundary to the
// right edge of the C partition (max), putting the free space within
// the extended partition. To Select an extended partition, the drive
// must first be selected, and then the partition.
Select Drive 1
Select Partition Extended
Resize Left Boundary Max

// Select the D partition and move it to the left, essentially
// flush against the Extended and C partitions, leaving the
// free space between the D and E partitions
Select Partition D
Move Left Max

// Select the E partition and move it as far as possible to the
// left, so that the free space will be at the end of E, within the
// extended partition
Select Partition E
Move Left Max

// The free space is now after E and the user
// can create an F partition (logical drive)

// Move to the free space after E
Select Partition Next

// Create the FAT partition called DATA with all defaults. This will
// use all of the size available in the free space.
Create /FS=FAT32 /Label="DATA"
```

Scenario 4 - Cluster Analyzer

The first physical drive is 30 GB in size and contains a 2047 MB primary FAT partition (C:) and an 8 GB hidden FAT32 primary partition. The drive also contains an extended partition with logical drives that uses up the remaining drive space.

The user would like to analyze the two FAT partitions to see if he can reduce the cluster waste.

```
// Show Cluster Waste for Partition 2
Select Drive 1
Select Partition 2
Cluster Analyzer /ShowClusterWaste

// Show Cluster Waste for Partition 3
Select Drive 1
Select Partition 1
Cluster Analyzer /ShowClusterWaste
```

If the cluster waste was too high on either partition we could use the Cluster Analyzer to reduce cluster waste. We will set the second partition to the recommended cluster size and then set the first partition to a cluster size of 8K.

```
// Set Partition 3 to Recommended Cluster Size
Select Drive 1
Select Partition 2
Cluster Analyzer /SetToRecommended

// Set Partition 2 to 8K Clusters
Select Drive 1
Select Partition 1
Cluster Analyzer /ClusterSize=8
```

Scenario 5 - Copy Partitions

The first physical drive is a 10 GB drive containing several primary and logical partitions. The second physical drive is a new, unformatted 60 GB drive. We would like to copy the first three partitions on drive 1 to drive 2.

```
// Select Drive 1, Partition 1
Select Drive 1
Select Partition 1

// Select Copy Drive 2, Copy Partition 1
Select Copy Drive 2
Select Copy Partition 1

// Copy First Partition
Copy

// Select Drive 1, Partition 2
Select Partition 2
// Select Copy Drive 2, Copy Partition 2
Select Copy Partition 2

// Copy Second Partition
Copy

// Select Drive 1, Partition 3
Select Partition 3
// Select Copy Drive 2, Copy Partition 3
Select Copy Partition 3

// Copy Third Partition
Copy
```

Scenario 6 - Partition Information

We have a 15GB primary C: partition that we would like to get information about. Specifically, we would like to know about the disk usage, cluster waste, partition information, and file system information.

```
// Select the first partition on the first drive and request
information
```

```
Select Drive 1
Select Partition 1
Info /Usage /Waste /Partition /FS

// This could also be accomplished using the alternate syntax: Select
Drive 1
Select Partition 1
Info /Usage
Info /Waste
Info /Partition
Info /FS
```

Scenario 7 - Delete Partitions

The second physical drive contains a 10GB primary NTFS D: partition labeled DATA and two 15GB logical partitions, E: and F:, labeled WORK and GAMES respectively . We would like to delete the E: and F: partitions then add the space to the D: partition.

```
// Make sure we do not accidentally delete any partitions on the
first physical
// drive
Select Drive 1
Set Drive Read Only Mode ON

// Select and delete the E: partition.
Select Drive 2
Select Partition E
Delete

// Select and delete the F: partition
Select Partition F
Delete

// Select and delete the extended partition
Select Partition Extended
Delete

// Now resize the D: partition to include the space previously used by E: and F:
Select Partition D
Resize Max
```

Scenario 8 - Delete All Partitions

The computer has three physical drives. The first physical drive is 40 GB in size and contains several primary and logical partitions. We would like to remove all partitions on the first drive so that we can reinstall the operating system.

```
// Make sure we do not accidentally delete any partitions on the
second or
// third physical drives
Select Drive 2
Set Drive Read Only Mode ON

Select Drive 3
```

```
Set Drive Read Only Mode ON
```

```
// Delete all partitions on the first physical drive
Select Drive 1
Delete ALL
```

Scenario 9 - Format Partition

The first physical drive is 60 GB in size and contains a 20GB primary NTFS partition (no volume label) and a 40 GB FAT32 logical partition. We want to reformat the primary NTFS partition as FAT32 and install a different operation system.

```
// Select the first partition then reformat it
Select Drive 1
Select Partition 1
Format /FS=FAT32 /Label="NEWOS"
```

Scenario 10 - Hide, Unhide, and Set Active

The first physical disk contains four primary partitions, each having a different operating system. All of the partitions are hidden except the third partition, which is the current active or boot partition. We want to hide the third partition and set the first partition as the active boot partition.

```
// Unhide the first partition then set it active
Select Drive 1
Select Partition 1
Unhide
Set Active
```

```
// Select the previous boot partition (third partition) and hide it
Select Partition 3
Hide
```

Scenario 11 - Show Partitions

The computer contains two physical drives but we are not sure what partitions are on the drive and are uncertain about drive lettering and partition IDs. We want to display a list of partitions for each drive.

```
// Show all partitions on the first physical drive
Select Drive 1
Show Partitions
```

```
// Show all partitions on the second physical drive
Select Drive 2
Show Partitions
```

If the /LOG or /DSS switches are used, the following table will be written to the log file or to the screen:

```
Volume          Type         Size MB  Used MB  Free MB  Status     Pri/Log
-------------- ------------ -------- -------- -------- ---------- -------
1 - *:WINNT4    Hid.FAT16B      902.1    399.1    503.0 Hidden     Primary
2 - *:WIN95C    FAT32           706.0    124.5    581.5 Active     Primary
3 - *:WINXP     Hid.FAT16B    2,039.5    622.0  1,417.5 Hidden     Primary
```

```
4 - *:          Hidden HPFS     549.1      7.2    541.9 Hidden     Primary
5 - *:          Free Space   15,366.9      0.0 15,366.9 None       Primary

Volume             Type         Size MB  Used MB  Free MB  Status     Pri/Log
---------------    ------------ -------- -------- -------- ---------- -------
1 - *:SWAPSPACE2   Linux Swap      141.2      0.0    141.2 None       Primary
2 - *:/            Linux Ext2    1,717.9    952.6    765.3 None       Primary
3 - *:             Extended      4,290.8  4,290.8      0.0 None       Primary
4 - *:TOOLS        FAT16B        1,921.8  1,552.8    369.0 None       Logical
5 - *:DATA         Hidden FAT32  2,368.9  1,698.2    670.8
```

CHAPTER 9

PQIDeploy

Overview

The Symantec imaging tool, PQIDeploy, is a special version of ImageCenter (formerly Drive Image Pro) designed to work in a configuration center or system builder production line. It contains the full ImageCenter scripting capability. It does not include any graphical user interface (GUI). All commands are specified in a script command file. Log and error files can be produced to verify correct operation.

This scripted mode is especially useful where many computers are being configured, as it automates the process and saves time by eliminating the need for user intervention.

The PQIDeploy system consists of two programs:

- PQIDplyD.exe (for DOS)
- PQIDeploy.exe (for Windows NT/2000/XP/PE)

When describing a feature common to all versions, the name PQIDeploy is used; when describing a feature specific to one version, the specific program name is used.

PQIDeploy scripting has been enhanced so that you can run the same scripts in ImageCenter or PQIDeploy without making modifications.

Running PQIDeploy

To run PQIDeploy, specify the program name followed by optional switches. For example:

```
PQIDPLYD {switch, [switch]...}(DOS)
PQIDEPLOY {switch, [switch]...}(Windows NT/2000/XP/PE)
```

PQIDeploy is run in *script* mode where the commands are specified in a script file and the command line specifies just the script file name and other switches.

Script files are text files that have one command on each line. Blank lines or lines that start with "//," "#," or ";" are ignored. Lines that start with "//," "#," or ";" can be used for comments within the script file. A sample script file is:

```
// Restore PQI to 2nd hard drive
SELECT DRIVE 2
DELETE ALL
SELECT FREESPACE FIRST
SELECT IMAGE ALL
RESTORE
```

Lack of GUI and Dialog Boxes

PQIDeploy does not have a GUI (graphical user interface). Instead it displays progress using text on the console. In cases where the GUI versions (Drive Image, ImageCenter, etc.) might display a dialog to ask a question, PQIDeploy prints the text from the dialog to the console. By default, it will then automatically answer the question, if the question has been configured with a default answer. All questions relating to fixing problems contain a default answer, which is to fix the problem. The only questions that don't have a default answer are those that really require a user response before it is possible to continue. For example, when PQIDeploy is restoring a PQI file that spans multiple CDs, it really can't go on until the user has changed the CD. In cases such as this, PQIDeploy does not automatically answer the question, but instead displays a prompt to the user, and waits for a response from the user. In this case, it would display a message similar to the following:

```
Please insert the next CD

Press <O> or <Enter> to proceed; Press <C> or <Escape> to cancel
```

The program then waits until one of the four keystrokes is pressed by the user.

Two switches are available to change this default behavior with regard to automatically answering questions. The /PROMPT command causes PQIDeploy to never automatically answer questions. When used, PQIDeploy will always display the question and then wait for the user's response. See "/PROMPT" on page 138 for more information.

The /NMD command causes PQIDeploy to always automatically answer questions. If a question does not have a default answer, PQIDeploy will use an answer of "Yes" or "OK." This was the default behavior of earlier versions of PQIDeploy, before the /NMD and /PROMPT switches were added. See "/NMD" on page 137 for more information.

Spanned PQI Image Files

Under Windows or DOS, PQIDeploy works with spanned images that are all stored in the same directory on a local drive or a network drive. PQIDeploy also works with spanned images on removable media (such as CD-ROM); PQIDeploy will prompt the user when the media needs to be changed.

PQIDeploy Boot Environment and Limitations

It is expected that PC configuration centers and production lines will employ one of three methods to boot the PCs before running PQIDeploy to set up their hard drives:

- Boot from a floppy (only possible with DOS)
- Boot from a CD
- PXE boot over the network

In all three cases, the operating system will not use the hard disk, so PQIDeploy has complete access to the drive and is able to modify it as needed.

It is also possible to run PQIDeploy when the computer has been booted from an OS installed on the hard disk, but there are limitations to the operations that PQIDeploy can perform in these instances. Both versions of PQIDeploy will work without problems on a blank second hard disk on the system or with free space at the end of the first hard disk. However, there are limitations to what they can do to partitions that are mounted and being used by the OS.

The DOS version, PQIDplyD, has the fewest restrictions. It can create images of all partitions and can even replace the boot/OS partition as long as the location from which it is accessing the PQI file is not modified. If the boot partition is modified, then PQIDeploy should reboot when finished instead of returning to the OS.

The Windows NT version, PQIDeploy, can only image or restore partitions that are not currently mounted by the OS. If a partition is mounted, PQIDeploy will attempt to unmount it, and if successful, will perform the operation. If it is unable to unmount the partition, then the operation will fail. Windows NT, Windows 2000, and Windows XP do not allow the boot partition to be unmounted, so PQIDeploy will never be able to image or modify these partitions. However, WinPE does allow the boot partition to be unmounted, although great care must be taken in the operations performed in this case.

Command Line Switches for PQIDeploy

As a minimum, the /CMD=<script-file> command line switch must be specified. Other command line switches are optional. For example: To specify SCRIPT.PQS as the script file, ERROR.TXT as the error file and E:\IMAGES\DRIVE2.PQI as the image file, enter the following:

```
PQIDPLYD /CMD=SCRIPT.PQS /ERR=ERROR.TXT /IMG=E:\IMAGES\DRIVE2.PQI
```

Under DOS, you are limited to 128 characters on the command line. You can overcome this limitation with the /ARG switch (see page 134). The limits under Windows are somewhat larger.

The DOS and Windows versions accept either a dash (-) or slash (/) before command line switches.

The following table lists the commands common to all versions of PQIDeploy.

Command Line Parameter	Description
/? or /H or /HELP	Displays all available command line switches, their syntax, and a short description.
/AFX	Automatically fix partition errors. Drive Image and ImageCenter detect various types of partition table errors, such as the CHS values not matching the LBA values, and so forth. If these programs can determine what the correct values should be, they either fix them automatically, or display a prompt asking if they should be fixed. PQIDeploy does not check for these errors or attempt to fix them, unless specifically requested to do so using this switch.
/ARG=<file>	Includes additional command line arguments in a text file. This is useful when the amount of data required for all the needed switches is too long to fit on a DOS command line, which is limited in length. It is also useful to consolidate commonly used sets of switches into a single location that many scripts can refer to, so that all scripts can be updated by changing just a single file. Each argument in the file must be preceded by either a “/” or “-” just as on the command line. Arguments within the file can be separated by a space or each can be on a separate line. Any line beginning with a “#” or “;” is ignored; they can be used for comments. The file can contain references to environment variables; they will be resolved the same as on the command line. This switch can be used multiple times on the command line or even from within a file specified by a /ARG on the command line.
/CAS	Copy All Sectors: Switch causes every sector within a partition to be saved and restored. This will include all information in a partition including the deleted files. This option significantly slows down the imaging process and increases file size.
/CEC	Check for an Extra Cylinder. This corrects a problem of the BIOS and DOS reporting different maximum cylinder numbers.
/CMD=<script-filename>	Designates a script command file; the script is a text file that contains arguments that are passed to the program.

Command Line Parameter	Description
/CMD=RESTORE_ALL	Deletes all current partitions on the specified drive and then restores all partitions in the image file (specified by the /IMG switch) to the drive. This command works in conjunction with the /DSK switch.
/CMD=STORE_ALL	This will store all partitions on the specified drive to the file specified by the /IMG= switch. This command works in conjunction with the /DSK switch.
/CMP=None | Low | High	Specify the compression level for storing images. This option overrides any compression level specified in the script.
/DBG[=<debug-filename>]	Specifies an output filename for the debug file. Symantec Technical Support can use the information found in the debug file to help diagnose problems you may encounter. If the <filename> is not specified, then PQ_DEBUG.TXT will be used. The DOS and Windows versions always create a debug file, even if the /DBG switch is not used. If you do not want a debug file automatically created, you can use the /NDBG switch on these platforms.
/DEL=<disknum>	Where <disknum> is either a disk number or an asterisk (*). If a number is specified, all partitions on that disk are deleted during application startup. If an asterisk is specified, all partitions on all disks are deleted during application startup. The Delete switch cannot be undone, and will be completed even if there is a syntax error in a script file (specified with the /CMD switch), or if the specified image file cannot be opened. The /DEL command does not wipe out the Master Boot Record. If you want to replace the MBR, either use the /MBR command to restore a MBR from an image file or use the /WFS command along with the DELETE ALL script command.
/DSK=<number>	Specifies disk number to use for restore or save operations. This option is only valid with /CMD=RESTORE_ALL or /CMD=STORE_ALL, or when PQIDeploy is being used as a PowerCast client.
/ERR=<error-filename>	Designates an error file. Because scripts execute without user intervention, these files are important for checking errors. If an error occurs, this file will contain only the error number.

Command Line Parameter	Description
/F4K	If the image was created of a partition with 4K (or less) cluster sizes, using this switch limits FAT32 to a 4K cluster size when restoring and resizing. If you do not use this switch, the cluster size will be resized up when the partition size becomes larger than 8 GB. Using this switch can significantly reduce restore times when resizing a FAT32 partition; however, you should only use it if you plan to convert the partition to NTFS later.
/IDE	(Integrated Drive Electronics – a standard used by hard drives to communicate with the controller ports or cards that allow the hard drive to interface with the computer) This switch is enabled by default. If you do not want it, you must specifically disable it by typing “/IDE=off” on the command line.
/IFC	Ignores file system checks. This allows a partition with a known file system error (for example, cross-linked files) to be stored in an image. Likewise, that same partition can be restored if this switch is used and if during the restore process the partition does not need to be resized.
/IHF	(Ignore hibernation files) Decreases the amount of time needed to start up ImageCenter on computers with many partitions by skipping checks to see if the machine was shut down while in hibernation.
/IMG=<image-filename>	Designates the path and filename for an image file. An alternative is to specify the image in the script file using the SET IMAGE FILENAME command. The STORE and RESTORE commands will encounter an error if an image file is not specified. If you are creating an image and specify an existing filename, the new image will replace the existing image. See “Accessing Hidden Partitions and Images Piped in on STDIN” on page 145 for details on some special uses of the /IMG switch. See “Reading PQI files from CDs in DOS” on page 142 for information about using this switch when reading CDs from DOS.
/INFO	List information about each partition, including the numbering used by scripts.
/LOG=<log-filename>	Designates a log file. Because scripts execute without user intervention, log files are important for checking results.
/MBI=[<disk-number>]	Image the MBR of the specified disk drive.
/MBR=[<disk-number>]	Restore the MBR to the specified disk drive.

Command Line Parameter	Description
/MFS=<span size>	Designates the maximum number of bytes, kilobytes, megabytes, or gigabytes that an individual image file can be. For example, the following are all equivalent: /MFS=1073741824 /MFS=1048576K /MFS=1024M /MFS=1G The minimum allowed value is 1048576, (and equivalents 1024K or 1M). When the specified size is reached, the image file is split. When running the DOS version, by default, images are spanned when they reach 2 GB in size, because this is the maximum file size supported by DOS.
/NMD	Forces PQIDeploy to never prompt for user input. All questions will automatically be answered with the default response if any or with “Yes” or “OK” if there is no default answer. See “Lack of GUI and Dialog Boxes” on page 132 for details.
/NRB	No Reboot. Suppresses the normal reboot during program exit. Causes the program to exit with an exit code of one if it would have rebooted without this switch and the program otherwise had no errors. Because the Windows version should never reboot the system, you do not need to use this switch under Windows.
/NUI	No User Interface. Don’t display progress or other information while the script is running.

Command Line Parameter	Description
/PCD=<IP-address>	Specifies the IP address that PowerCast clients should use to discover their PowerCast server. IP address is of the form W.X.Y.Z, where X, Y and Z must be in the range 0-255, and W must be in the range 224-239. The default address is 224.47.0.1. All of the PowerCast clients and server must use the same IP address for discovery. If multiple PowerCast servers are running, they can and should use the same IP address for discovery, but they must each have a unique session name (/MSN) and unique PowerCast session address (/PCS). It is recommended that if you specify the discovery address to the server, you also specify the session address (/PCS). If you specify the /PCD address to the server, you MUST use the /PCD option with all of the clients, specifying this same address.
/PGS=<file>	Creates the specified file and writes all the progress information to it as the program runs. This switch is designed so that if a file is written to a network share, then someone could remotely monitor the progress of PQIDeploy. Note that only progress information is written to this file. If you need error information, look in the error file or the log file.
/PPR=<partition-list>	Protect Partitions. Before PowerCasting begins, the client deletes all partitions on the target disk. This option allows the user to protect the specified partitions from being deleted. <partition-list> is a comma separated list that supports specifying of partitions by number, drive letter (without a colon following), or the words FIRST, LAST, DIAGNOSTIC, UNKNOWN, or ALL. (server only) See also "/PPR=<partition-list>" on page 141.
/PROMPT	Forces PQIDeploy to always prompt for user input. Normally, PQIDeploy automatically answers question that have a default response, such as whether the application should fix partition errors that it has found. (See "/AFX" on page 134.) The default answer is to fix all problems encountered. Using the /PROMPT switch allows the user to make these decisions themselves. See "Lack of GUI and Dialog Boxes" on page 132.

Command Line Parameter	Description
/PWD=<password>	Designates a password that must be given to restore the partition's data. The password may be either a number or a string. This switch is used when creating a new PQI that should be encrypted. The switch must also be used to restore a PQI that was created with encryption. The password can also be specified in a script file using the SET PASSWORD script command. For restoring, it is also possible to embed the password in the RTC file so that is it not visible to the end user.
/RAV	Read and Verify Disk Writes: Causes each sector restored to disk to be read back and compared as an extra security precaution. Using this parameter will significantly increase restore time.
/RPP	Resize partitions proportionally during restore operations.
/SCO	Syntax Check Only. Causes the syntax of each command in the script file to be checked without executing the command. This is useful when writing and debugging the script file.
/SNC	Scripting No Cancel. Disable <Ctrl-C> so that a script cannot be stopped by pressing <Ctrl-C>. Normally, if you press <Ctrl-C>, PQIDeploy will abort gracefully, stopping within a few seconds and only after it has cleaned up the partition table.
/SSO	Store system structures only. This creates an image file that contains only the file system structures and no file data.
/TLG	Truncates the log file so that all previous log entries are deleted. This cleans out the log file so that only information from the last run of the program is contained in the log. Without this switch, log information is appended to an existing log file.
/VER	Displays information about the version of PQIDeploy you are running.
/VIP	Verify Image Partitions. When creating an image, after the image creation is complete, verify that the contents of the image match the contents of the disk that was just imaged. Using this option doubles the time it takes to create an image, since it requires two complete passes of the disk contents, one for the original creation of the image and the second to ensure that it matches the image.

Command Line Parameter	Description
/WFS	Wipes the first sector after deleting all partitions. This will insure that a fresh master boot record will be written to the destination disk.
/ZLB	Zlib Compression. When creating an image, use Zlib compression instead of the standard PQ Compression algorithm. Creating images with Zlib compression generally takes longer, but produces smaller images. Restore times may be faster though, since less data needs to be read when reading the PQI file.

PowerCasting Client

PQIDeploy can serve as a PowerCast client. However, only the DOS and NT versions of PQIDeploy support PowerCasting. The following PowerCast Client command line parameters are supported in the DOS and Windows NT versions of PQIDeploy.

Command Line Parameter	Description
/HPC	Hides partitions after copying (restoring) them.
/MSN=<session-name>	Multicast session name to use for this PowerCast session. The PowerCast Server and all of the clients must specify the same multicast session name. When using the /MSN switch, the /CMD and /IMG switches should never be used. The following switches will be ignored when the /MSN switch is used: CAS, CMP, MBI, MBR, MFS, SCO, SSO, VIP, WFS, and ZLB.

Command Line Parameter	Description
/PPR=<partition-list>	This switch provides Partition Protection when PowerCasting. Normally, PowerCasting deletes all partitions on the specified drive, but this switch can be used to protect selected partitions from being deleted. <partition-list> is a comma separated list of partitions. No white space is allowed in the list unless the whole list is in double quotes. All text is case-insensitive. Partitions can be specified using any of the following forms: *partition number* - The partition to protect by number. *drive letter* - The partition to protect by drive letter (single character without colon). Drive letters may be upper or lower case. This refers to the drive letter used by the current OS. If PQIDeploy is running in DOS, the drive letters will likely be different from the letters used by Windows. FIRST - Protect the first partition. LAST - Protect the last partition. DIAGNOSTIC - Protect a valid partition of an unknown type that is either the first or the last non-free space partition. UNKNOWN - Protect all valid partitions of an unknown type. ALL - Protect all partitions. There must be enough free space available where the image can be restored. This overrides all other forms of *<partition-list>*.
/TTL=<routers>	Sets the Time to Live value to specify the number of routers a PowerCast session can pass through before it is killed. Enter a value between 1 and 32. Default is 1. Both clients and the server may need to specify this value, and the server value should always be at least as large as the largest client value. If some clients are known to be closer to the server, they may use a smaller value though. Use the smallest value that works to minimize the network load.

Use the following command line to start PQIDeploy as a PowerCast client:

```
PQIDeploy /MSN="session name" /DSK=1 /PPR=1,2
```

This will start PowerCasting in client mode, attach to the specified PowerCast session, delete all partitions on the specified disk that are not protected with the PPR switch, and restore the partitions from the image file received from the PowerCast server.

When a PowerCast session is completed, PQIDeploy will always reboot the system, unless the /NRB (no reboot) switch is specified on the command line.

If you are using PQIDeploy in conjunction with Symantec DeployCenter, you can substitute the DOS version of PQIDeploy for the DOS PowerCast client (PQPCAST.EXE) on the PowerCast boot diskette. For more information about PowerCasting, refer to Chapter 6 of the Symantec DeployCenter user guide.

Commands Specific to the DOS Version (PQIDplyD)

The following table lists the switches specific to the DOS version of PQIDeploy (PQIDplyD).

Command Line Parameter	Description
/IDE=ON | OFF	When ON, use UDMA instead of INT13 when talking to an IDE disk controller. For most controllers this will significantly improve performance. However, it may not work correctly on all IDE controllers, especially on very old computers. It has no effect if the disk being used for the save or restore is not attached to an IDE controller.
/OVI=0 | 1	Overlapped I/O. This switch should only be used when /IDE=ON is specified. This switch only affects image restore. It does not affect image creation. When /IDE is specified, PQIDeploy will attempt to determine whether or not overlapping the I/O (Reading the PQI and writing the disk sectors) can be safely and efficiently overlapped. If you want to override the default, use this switch. 0 disables overlapping while 1 enables it. Overlapping should not be enabled if the PQI file is on the same disk as the partition being restored, or if the PQI is on a second disk that uses the same IDE channel as the disk being restored. Overlapping, when appropriate, will generally improve performance, except when the image being restored is highly or moderately fragmented.
/NDBG	The DOS version of PQIDeploy always creates a debug file, unless /NDBG is specified.

Reading PQI files from CDs in DOS

Normally, DOS requires the user to load several device drivers in the CONFIG.SYS and AUTOEXEC.BAT files in order for DOS programs to be able to access data on a CD. These device drivers assign a drive letter to the CD-ROM drive. These drivers are often proprietary, and there may be restrictions on redistributing them. Many drivers are also many years old, unsupported by the creators, and not very reliable. For these reasons, the DOS version of

PQIDeploy has built in support for reading PQI files from CD, without the need for the user to load any special device drivers. The built in support is generally faster and more reliable than the support provided by DOS device drivers.

To access a PQI file from a CD, use a filename of **\\.\pqcd#\file.pqi**, where # is the CD-ROM device number (generally 0), and **file.pqi** is the path/name of the PQI file on the CD. For example, the following will read the PQI “\PQ\IMAGES\WINXP.PQI” from the first CD-ROM drive on the system:

```
PQIDPLYD /IMG=\\.\pqcd0\pq\images\winxp.pqi ...
```

If the PQI spans multiple CDs, the subsequent portions of the PQI (winxp.002, winxp.003, etc.) should be in a directory on the subsequent CD that has the same name as the directory on the first CD. PQIDeploy will automatically prompt the user to change CDs if the image spans multiple CDs.

Commands Specific to the Windows NT Version (PQIDeploy)

The following table lists the commands specific to the Windows NT/2000/XP/PE version PQIDeploy.EXE.

Command Line Parameter	Description
/NSU	No System Update. Normally, PQIDeploy for Windows NT notifies the OS after it restores partitions or otherwise modifies the partition table. This notification allows NT/2000/XP/PE to assign drive letters to the new or modified partitions, so that other programs can immediately access these partitions without requiring a reboot. However, in some cases, the notification handshake can take up to a couple of minutes. If you plan to reboot before accessing these modified partitions, you can use the /NSU switch to skip the notification step and save time. Note that the system update is not always this slow; in many cases, the process is complete in just a few seconds. The long delay generally happens when PQIDeploy deletes some number of partitions and then immediately restores the same number of partitions. One other option in this case is to use DISKPART.EXE or some other Windows NT utility to delete the partitions first and then use PQIDeploy only for the restore.

Command Line Parameter	**Description**
/OVI=<number>	Overlapped I/O. Controls the multi-threading used by this version. This switch is only used when restoring an image. 0 - no multi-threading. 1 - Overlap PQI reading and decompression with Disk writes (2 threads) 2 - Overlap PQI reading with decompression (2 threads) 3 - Overlap PQI Reading with decompression and with Disk writes (3 threads) 4 - Same as 0 5 - Same as 1 6 - Same as 2 but with PQI read-ahead 7 - Same as 3 but with PQI read-ahead The default value is 3. Setting this value optimally can significantly improve performance. When the PQI is on the same disk as the partition that is being restored, the optimal value will probably be either 2 or 6. In most other cases, the optimal value will be 3 or 7, unless the partition data is highly fragmented.
/PRI=<number>	Set process priority. Accepts values 1-6. 1 - Idle priority 2 - Below normal priority (New to XP - not yet supported) 3 - Normal priority 4 - Above normal priority (New to XP - not yet supported) 5 - High priority 6 - Real-time priority The default value is 3. If no other processes are running on the machine at the same time as PQIDeploy, this switch has very little effect. However, when other processes are running, such as system diagnostics, you can use it to control whether PQIDeploy runs at a higher or lower priority than those other processes.

Command Line Parameter	Description
/NDBG	The Windows version of PQIDeploy always creates a debug file, unless /NDBG is specified.

Environment Variables

There are two environment variables that can be used for performance tuning of PQIDeploy: PQ_IMAGEBUFFERSIZE and PQ_TRANSFER.

PQ_IMAGEBUFFERSIZE controls the size of an internal buffer used by PQIDeploy when saving and restoring images. On DOS and NT this buffer defaults to half of available physical memory. On all platforms, you can specify a size to use, overriding the default, as long as there is sufficient available memory. The size should be at least 256K. The number will be rounded to the nearest multiple of 64K. When setting this environment variable specify the number of kilobytes or megabytes, followed by the letter K or M respectively. For example, on DOS or Windows:

```
set PQ_IMAGEBUFFERSIZE=8M
```

will set the buffer size to 8 MB. For the DOS version, this variable should not be set when overlapping I/O is enabled. (See “/OVI=0 | 1” on page 142 for PQIDplyD.)

PQ_TRANSFER is the second environment variable and is only used with the DOS version. PQ_TRANSFER can be set to the numbers 0 to 15. This variable relates to the packet sizes used when reading or writing the PQI file, especially over the network. If not specified, the default value is 2, which in Symantec testing is usually the optimal value. However, in some networking environments, where PQIDeploy is performing poorly, other values have been found to more than double the performance. If performance is poor, setting PQ_TRANSFER to 15 will sometimes dramatically improve performance. In order of likelihood, the four values most likely to provide optimal performance are 2, 15, 0, and 6. Note that PQ_TRANSER is not used by the Windows version of PQIDeploy.

Performance Tuning

The /IDE, /OVI and /PRI switches can be used along with the environment variables described above to optimize the performance of the various versions of PQIDeploy for a particular environment. Symantec has also added performance information to the debug file created with the /DBG switch. Contact your Symantec sales engineer or Symantec Professional Services for help in interpreting this information and in maximizing the performance of PQIDeploy on your systems.

Accessing Hidden Partitions and Images Piped in on STDIN

PQIDeploy has the ability to read images from and write images to hidden partitions as well as reading images from STDIN (its standard input stream). A hidden partition is any partition not currently mounted by the file system. Under DOS and Windows, this generally means that the partition is not assigned a drive letter. Both of these variants for accessing images are supported through special syntax via the /IMG command line switch or the SET IMAGE FILENAME command in the script.

To store an image in an otherwise inaccessible partition (type FAT16, FAT32, or NTFS only) invoke PQIDeploy with the following image tag, where the disk and partition number are discoverable using the /INFO command line option.

```
/IMG=\\.\disk1.part2\path\image.pqi
```

You would also substitute the appropriate path and file name for *path* and *image.pqi* in the above example. You would also change the *disk* and *part* numbers to access the desired disk and partition. The numbers appearing after the *disk* and *part* portions are the same numbers that would be used in a script with the SELECT DRIVE (page 168) and SELECT PARTITION (page 169) commands. You can use the /INFO command line switch to get a listing of the assigned disk and partition numbers.

The /INFO command also specifies PQAccess numbers for each partition on all disks, which consists of a number followed by a colon. PQAccess syntax can be substituted for the '\\.\disk1.part2' portion of the above example, as follows:

```
/IMG=2:\path\image.pqi
```

The above syntax can be used for both reading and writing PQI files on hidden partitions. The Windows version of PQIDeploy also allow them to read the PQI file from STDIN.

```
/IMG=\\.\special\stdin
```

With this included in the command line an image can be passed via a redirect (<), a pipe (|) or a named pipe. For example:

```
pqideploy /img=//./special/stdin /cmd=script.pqs < image.pqi
```

The above example provides no benefit over specifying the name of the PQI file directly to the /IMG command, and in fact may slightly degrade performance. This syntax is more likely to be used with a pipe or named pipe to access a PQI that is not stored locally, or stored in a proprietary way. In this case, another application would access the PQI file and pass the data into the pipe for use by PQIDeploy. Note that DOS does not support the STDIN functionality.

PQIDeploy Scripting

The script file designated on the command line contains arguments or instructions that are passed to the program, determining which operations are executed.

Because the script files execute without user intervention, use extra care when developing the script file. For example, if the DELETE ALL command is encountered, all the partitions on the currently selected drive will be deleted without any warning or confirmation messages to cancel the operation.

Variables can be used anywhere on a PQIDeploy script line and mathematical expressions can be evaluated with the results assigned to local variables or used as parameters to commands. If a local variable is not defined, but an environment variable with the same name exists, then the value of the environment variable will be used.

Data Types

The following data types are supported with PQIDeploy scripting.

Data Type	Example	Description

<integer>	`10000`
	`0x13FE`
		A 32-bit value. Any whole number between about -2,000,000,000 and 2,000,000,000. An <integer> cannot have a decimal point or commas. An <integer> can be specified in hexadecimal format by preceding the value with 0x.

<number>	`1`
	`3.15159`
		Any real number. All <integers> are <numbers>, but many <numbers> are not <integers>. For example, 3.2 is a <number> but not an <integer>. Any place a <number> can be used, an <integer> can also be used, but you cannot substitute an <number> where an <integer> is required.

<string>	`"Hello"` `'Hello'`	Any text string. Strings should generally be enclosed in quotes. You can use single or double quotes. If you want to include a quote character within a string, use the opposite type of quote character around the string, as in: `'"Hello,"` she said.'

Data Type	Example	Description

<var>	`%BAT%`	A local variable. At startup, each environment variable is imported as a local variable with the same name and value. However, if a local variable is modified, its corresponding environment variable is not modified. Local variables are used similarly to how environment variables are used in a DOS batch file. Variables are not evaluated if they are enclosed in quotes. However if you want to have the result of a variable evaluation treated as a string, then you can surround it by a backslash then a quote. For example, assuming %var% evaluates to
		`Hello World`
		then
		`\"%var%\"`
		evaluates to
		`"Hello World"`
		If a local variable is undefined, then no substitution is made in the text; instead the variable name along with the percent symbols is left in the text. For example, using some of the operators defined below, these two expressions are equivalent:
		`IF \"%var%\" = "%var%" GOTO NotDefined`
		`IF NOT DEFINED var GOTO NotDefined`

<Boolean>	`true` `false`	Either TRUE or FALSE. The words "true" and "false," without quotes, can be used in a script to represent these values. If they are enclosed in quotes, then they are strings.

Operators

The table below describes the supported mathematical and logical operators. Unless otherwise noted, all operators are valid for operands of type <integer> and <number> only. If all operands are the same type, then the result is that same type unless otherwise noted. If <integer> and <number> operands are both used, result is a <number> unless otherwise noted.

Operator	Example	Description

-	`-1`
	`-(5*3) = -15`
		Unary negation. Change the sign of the value to the right of the minus sign.

~	`~0x13fc = 0xffffec03`	Bitwise negation. Invert the value of each bit in the <integer>. If you want to display the result of ~0x13fc in hexadecimal, use the function HEX (~0x13fc) which results in 0xffffec03. This value is equivalent to -5117 decimal.
**	`2**10 = 1024`	Power Operator. Raise the value to the left to the power of the value to the right.
*	`2 * 3 = 6`	Multiplication.
/	`5 / 2 = 2.5`	Division. Result is always a <number>. Use "INT (5 / 2)" if you want an <integer> result.

%	`5 % 2 = 1`
	`5.4 % 2.1 = 1.2`
		Remainder. x % y = x – (INT (x / y) * y). In the examples, 5 / 2 equals 2 with a remainder of 1. 5.4 / 2.1 equals 2 with a remainder of 1.2

+	`2+2 = 4`
	`"test"+ "data"= "testdata"`
	`"test" + 8 = "test8"`
	`8 + "test" = "8test"`
		Addition. Add two numbers or concatenate two strings or concatenate a single string and a number.

-	`4 - 2 = 0`	Subtraction
<<	`2 << 2 = 8`	Left Shift the bits of an <integer>.
>>	`8 >> 2 = 2`	Right shift the bits of an <integer>.

Operator	Example	Description
&	`0xE5 & 0x7F = 0x65`	Bitwise AND operator. Valid only for <integers>.
|	`0xE5 | 0x7F = 0xFF`	Bitwise OR operator. Valid only for <integers>.
^	`0xE5 ^ 0x7F = 0x9A`	Bitwise XOR operator. Valid only for <integers>.
:=	`set var greeting:= "hello"`	Assignment operator, used with the commands SET VAR, GET INI and GET REG. The value to the right of the assignment operator can have any data type.

= ==	`"hello" == "HELLO"`
	`10 = 2 * 5`
		Comparison operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive, so `"hello"` is equal to `"Hello"`. Result is a <Boolean>.

!= <>	`"hello" <> "world"`
	`11 != 2.5`
		Inequality operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive. Gives the opposite result from = or ==. X <> Y is equivalent to NOT (X == Y). Result is a <Boolean>.

<	`"a" < "b"`
	`0.5 < 1`
		Less than operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive. Result is a <Boolean>.

>	`"aba" > "aac"`	Greater than operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive. Result is a <Boolean>.

<=	`"a" <= "a"`
	`0.5 <= 1`
		Less than or equal operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive. Result is a <Boolean>.

>=	`"aaa" >= "aaa"`	Greater than or equal operator, used to compare <strings> to other <strings> or <numbers> to other <numbers>. For strings, the comparison is case insensitive. Result is a <Boolean>.

Operator	Example	Description
NOT	`(NOT TRUE) = FALSE`	Invert the value of a <Boolean> or Boolean expression.
AND	`(TRUE AND FALSE)= FALSE`	A <Boolean> that is only true if both of its operands are true.
OR	`(TRUE OR FALSE)= TRUE`	A <Boolean> that is true if either of its operands are true.
XOR	`(TRUE XOR FALSE)= TRUE`	A <Boolean> that is only true if one of its operands is true and the other is false.

Operator Precedence

The following table describes the associativity and operator precedence of the operators listed in the previous section. Items on the same line have equal precedence, and precedence decreases as you go down. Parentheses can be used around any operator to force evaluation in the desired order. For example, (5 + 7) * 1024 adds five and seven before multiplying the result by 1024. Without the parentheses, the result would be different.

Operators	Associativity
- (negation) ~	Right to Left
**	Left to Right
* / %	Left to Right
+ - (subtraction)	Left to Right
<< >>	Left to Right
&	Left to Right
^	Left to Right
|	Left to Right
= == != <> < > <= >=	Left to Right
NOT	Right to Left
AND	Left to Right
XOR	Left to Right
OR	Left to Right

Functions

Functions take operands that follow the function name. The operands must be enclosed in parenthesis, unless otherwise noted. All function names are not case-sensitive, so SQRT, sqrt, and sQrT are all equivalent.

Function	Operand Type	Example	Description
ABS	<number>	`ABS (-3) = 3`	Absolute value of operand. The result is never a negative number.
SQRT	<number>	`SQRT (9) = 3`	Square root of operand. Operand must be non-negative. Result is always a <number>, never an <integer>.

INT	<number>	`INT (6.9) = 6`
		`INT (-6.9) = -6`
			Integer portion of a number. Result is always an <integer>. Result is obtained by truncating any value after the decimal point.

RND	<number>	`RND (6.9) = 7`
		`RND (-6.9) = -7`
		`RND (6.1) = 6`
			Round number to the <integer> that is closest in value.

HEX	<integer>	`HEX (43) = "0x0000002B"`	Convert the <integer> to a hexadecimal <string>. Result is always a <string> starting with "0x" followed by an eight-digit value.

FMIN	<number>, <number>
	or
	<string>, <string>
		`FMIN (3,2) = 2`
		`FMIN("apple","orange")= "apple"`
			Result is the lower <number> of the two <numbers> or the lesser <string> of two <strings>.

FMAX	<number>, <number>
	or
	<string>, <string>
		`FMAX (3,2) = 3`
		`FMAX("apple","orange")= "orange"`
			Result is the higher <number> of the two <numbers> or greater of the two <strings>.

Function	Operand Type	Example	Description

IsNumber	<number> or <string> or <Boolean>	`IsNumber (%var%)`	Result is a Boolean. It will be true if input parameter has a type of <integer> or <number>. Otherwise it will be false. Note that IsNumber ("15") is false, because the value is inside of quotes and is therefore a string, not a number. If you have a string that you want to make into a number try
			`SET VAR NUMBER := <string>`
			Where <string> could be an expression that results in a string type. %NUMBER% will then be a <number>. If you have a number that you want to make into a string, use:
			`"" + <number>`
			where <number> could be a numerical expression or a variable.

StrLen	<string>	`StrLen ("Hello") = 5`	Returns the number of single-byte characters in a string. If a string includes UTF8 characters, the result will indicate the number of bytes, not the number of characters. Result is an <integer>.

Function	Operand Type	Example	Description
EXISTS	<string>	`EXISTS "c:\boot.ini"`	Checks if the specified file exists. Result is a <Boolean>. Parentheses around the file name are optional, but it should be enclosed in quotes. The file name can use a PQAccess drive number instead of a drive letter. For example, "1:\boot.ini" would refer the file boot.ini on the first partition of the first disk. Use the /INFO command to obtain the PQAccess drive numbers. To be compatible with DOS, you can also use the word EXIST instead of EXISTS as the name of this command.
FILESIZE	<string>	`FILESIZE "c:\boot.ini"`	Returns the size of the specified file. Result is an <integer>. Parentheses around the file name are optional, but it should be enclosed in quotes. The file name can use a PQAccess drive number instead of a drive letter. For example, "1:\boot.ini" would refer the file BOOT.INI on the first partition of the first disk. Use the /INFO command to obtain the PQAccess drive numbers. It is an error to call FILESIZE on a file that does not exist.
DEFINED	<string>	`DEFINED var`	Returns true if the specified variable (specified without using percent signs around it) has a value. Result is a <Boolean>.

Queries

Queries let you obtain various values related to disks and partitions and use them in expressions or other calculations.

Multi-Disk Queries

These functions do not take any parameters. Function names can optionally be followed by an empty set of parentheses to make it more obvious that this is a function call.

Example

```
SET VAR total_disks := GetTotalDisks()
```

Function	Return Type	Description
GetTotalDisks	<integer>	Returns the number of disks in the system.
QueryLargestUnusedSize	<number>	Searches all disks and finds the partition that has the most unused space in it. Returns the size of this space in megabytes. Returns 0 if there are no partitions on any disk or if they are all completely full.
QueryLargestUnusedDisk	<integer>	Searches all disks and finds the partition that has the most unused space in it. Returns the number of the disk that contains this partition. Returns 0 if there are no partitions or they are all completely full.
QueryLargestUnallocatedSize	<number>	Searches all disks and finds the largest free space. Returns the size of this space in megabytes. Returns 0 if there are no unallocated or free spaces on any of the disks.
QueryLargestUnallocatedDisk	<integer>	Searches all disks and finds the largest free space. Returns the number of the disk that contains this free space. Returns 0 if there are no unallocated or free spaces on any of the disks.

Disk Queries - Currently Selected Disk

These queries use the disk selected with the last SELECT DISK command. These functions do not take any parameters. Function names can optionally be followed by an empty set of parentheses.

Example

```
SELECT DISK 2
```

```
SET VAR disk_size := GetDiskSize()
```

Function	Return Type	Description
GetCurrentDiskNum	<integer>	Returns the number of the last disk specified when using the SELECT DISK command. Returns 0 if SELECT DISK has not been called.
GetDiskSize	<number>	Returns the size of the disk in megabytes.
GetTotalPartitions	<integer>	Gets the total number of partitions on the disk
GetAllocatedSize	<number>	Gets the accumulated size of all of the allocated partitions on the disk.
GetAllocatedPercent	<number>	Returns the percentage of the disk that is allocated. Result will be a number between 0 and 100.
GetTotalUnallocatedSpaces	<integer>	Returns the number of free spaces on the disk.
GetUnallocatedSize	<number>	Returns the accumulated size of all of the free spaces on the disk.
GetUnallocatedPercent	<number>	Returns the percentage of the disk that is unallocated. Result will be a number between 0 and 100.

Partition Queries

These queries use the partition selected with the last SELECT PARTITION command. These functions do not take any parameters. Function names can optionally be followed by an empty set of parentheses.

Function	Return Type	Description
GetCurrentPartNum	<integer>	Returns the number of the last partition specified when using the SELECT PARTITION command. Returns 0 if there is not currently a selected partition.
GetSelectedPartitionSize	<number>	Returns the size of the partition in megabytes.
GetUsedAmount	<number>	Returns the size of the used space on the partition in megabytes.

Function	Return Type	Description
GetUsedPercent	<number>	Returns the percentage of used space on the partition. Result will be a number between 0 and 100.
GetUnusedAmount	<number>	Returns the size of the unused space on the partition in megabytes.
GetUnusedPercent	<number>	Returns the percentage of unused space on the partition. Result will be a number between 0 and 100.
GetPartitionLabel	<string>	Gets the volume label of the selected partition. If the partition does not have a label, the result will be the empty string. If you assign this value (empty string) to a variable the result will be undefining the variable.
IsFat	<Boolean>	Returns true if the partition type is FAT16.
IsFat32	<Boolean>	Returns true if the partition type is FAT32.
IsNTFS	<Boolean>	Returns true if the partition type is NTFS.
IsHPFS	<Boolean>	Returns true if the partition type is HPFS.
IsLinuxExt2	<Boolean>	Returns true if the partition type is Linux Ext2.
IsLinuxExt3	<Boolean>	Returns true if the partition type is Linux Ext3.
IsLinuxSwap	<Boolean>	Returns true if the partition type is Linux swap.
IsActive	<Boolean>	Returns true if the partition is active (the selected boot partition for this disk).
IsHidden	<Boolean>	Returns true if the partition is hidden.
IsPrimary	<Boolean>	Returns true if the partition is a primary partition.
IsLogical	<Boolean>	Returns true if the partition is a logical partition, located within an extended partition.

Free Space Query

This query uses the free space selected with the last SELECT FREESPACE command. This function does not take any parameters.

Function	Return Type	Description
GetCurrentFreespaceNum	<integer>	Returns the number of the currently select free space. Returns 0 is there is no currently selected free space.

Disk Queries - Specified Disk

Each of these queries takes a disk number as a single argument. The disk number must be an <integer> and enclosed in parentheses. Specifying a non-existent disk number will result in an error.

Example

```
SET VAR disk2_size := QueryDiskSize(2)
```

Function	Return Type	Description
QueryDiskSize	<number>	Size of disk in megabytes.
QueryDiskNumPartitions	<integer>	Total number of partitions.
QueryDiskNumFreeSpaces	<integer>	Total number of free spaces.
QueryDiskNumPrimaries	<integer>	Total number of primary partitions (including the extended partition, if there is one). The maximum allowed number of primary partitions is four.
QueryDiskTotalAllocated	<number>	Accumulated size of all partitions in megabytes.
QueryDiskTotalUnallocated	<number>	Accumulated size of all free spaces in megabytes.
QueryDiskLargestUnusedSize	<number>	Searches all partitions and finds the partition that has the most unused space in it. Returns the size of this space in megabytes. Returns 0 if there are no partitions on this disk or if they are all completely full.
QueryDiskLargestUnusedPart	<integer>	Searches all partitions and finds the partition that has the most unused space in it. Returns the number of this partition. Returns 0 if there are no partitions on this disk or if they are all completely full.

Function	Return Type	Description
QueryDiskLargestUnallocatedSize	<number>	Finds the largest free space and returns the size of this space in megabytes. Returns 0 if there are no free spaces on this disk.
QueryDiskLargestUnallocatedFree	<integer>	Finds the largest free space and returns the number of this free space. Returns 0 if there are no free spaces on this disk.

Partition Queries - Specified Disk

These functions take two arguments: a disk number and a partition number. Both numbers must be <integers>, separated by a comma and enclosed in parentheses. Specifying a non-existent disk or partition will result in an error.

Example

```
SET VAR disk2_part3_size := QueryPartSize(2,3)
```

Function	Return Type	Description
QueryPartSize	<number>	Size of partition in megabytes.
QueryPartUsed	<number>	Size of the used space in megabytes.
QueryPartUnused	<number>	Size of the unused space in megabytes.
QueryPartType	<string>	Partition type as a string. The most common partition types are FAT, FAT32, NTFS, Ext2, and Ext3.
QueryPartLabel	<string>	Partition volume label returned as a string. If the partition does not have a label, the result will be the empty string. If you assign this value (empty string) to a variable the result will be undefining the variable.
QueryPartIsLogical	<Boolean>	True if partition is inside an extended partition.
QueryPartIsActive	<Boolean>	True if partition is the current boot partition on this disk.
QueryPartIsHidden	<Boolean>	True if the partition has a type that is considered hidden. Hidden partitions are not seen by DOS or Win9x but can be seen by Windows 2000 and Windows XP.

Free Space Queries - Specified Disk and Free Space

These functions take two arguments: a disk number and a free space number. Both numbers must be <integers>, separated by a comma and enclosed in parentheses. Specifying a non-existent disk or free space will result in an error.

Function	Return Type	Description
QueryFreeSize	<number>	Size of the free space in megabytes.
QueryFreeIsLogical	<Boolean>	True if the free space is located inside an extended partition.
QueryFreeCanBeLogical	<Boolean>	True if any of the following are true: • Free space is located inside of an extended partition. • Free space is adjacent to an extended partition, so the extended partition could be expanded to include the freespace. • There is currently no extended partition on the disk, and there are fewer than four primary partitions on the disk, so an extended partition could be created in this free space.

Scripting Commands

The following table lists the scripting commands used with PQIDeploy.

Script Argument	Action
APPEND “*<src-file>*” “*<dest-file>*”	Append a file to the end of an existing file. <src-file> will be appended to the end of *<dest-file>*. The *<dest-file>* should already exist. Both *<src-file>* and *<dest-file>* should be placed in quotes. The source file, destination file, or both can be on a hidden partition. To specify access to a hidden partition, use the PQAccess partition number displayed when using the /INFO command line switch to PQIDeploy. For example: `APPEND "Z:\FILE.002" "1:\images\image.pqi"` will append the file FILE.002 on Z: (probably a CD or network drive) to the end of \images\image.pqi located on partition 1: (the first partition on the first drive).
COPY “*<src-file>*” “*<dest-file>*”	Copies a single file, *<src-file>*, to a the location and name specified by *<dest-file>*. The *<src-file>* cannot contain wildcard characters. The *<dest-file>* must specify a complete path and filename. Both *<src-file>* and *<dest-file>* should be placed in quotes. The source file, the destination file, or both can be on a hidden partition. To specify access to a hidden partition, use the PQAccess partition number displayed when using the /INFO command line switch to PQIDeploy. For example: `COPY "Z:\network.sys" "1:\windows\network.sys"` will copy the file network.sys from Z: (probably a CD or network drive) to partition 1: (the first partition on the first drive).

Script Argument	Action
CREATE /FS <fs> [/LABEL="<label>"] [/SIZE=<size>] [/POSITION= <position>]	Create a new, empty partition in the currently selected freespace. /FS: <fs> should be set to one of {FAT | FAT32 | NTFS | LINUXEXT2 | LINUXEXT3 | LINUXSWAP | EXTENDED | UNFORMATTED}. /LABEL: The volume label is optional. If a volume label is desired, it should be placed between quotes. It must be less than 12 character for FAT/FAT32 partitions, less than 17 characters for Ext2 or Ext3 partitions and less than 25 character for NTFS. /SIZE: Size is specified in megabytes and will default to the full size of the selected free space if not specified. It can include a decimal point for fractions of a megabyte, but the size may be rounded up if necessary to align the end on a cylinder boundary. /POSITION: If the new partition does not fill all available free space, the position parameter can be used to specify whether the partition should be created at the beginning or the end of the free space. <position> should have a value of {BEGINNING | END}. If a position is not specified, the new partition will be placed at the beginning of the free space. Example: `CREATE /FS=FAT32 /LABEL=“Data” /SIZE=500`
DELETE [ALL | EXTENDED]	If no parameter is specified, this command deletes the last partition or partitions selected using the SELECT PARTITION command. ALL: Deletes all partitions on the currently selected drive, even if they are not selected. EXTENDED: Deletes the extended partition; the extended partition can only be deleted after all the logical partitions within it have been deleted.

Script Argument	Action
HIDE	Hides the currently selected partition.
MESSAGE [/<button> [= [GOTO] <label>]] [/<button> = ABORT] "<text>"	Displays the specified message <text> to the user. Example: `Message "There are" GetTotalDisks "drives in this machine."` For international users, the text can be in UTF8 format. If the /NMD switch is used, then the user will never see the message, otherwise the <text> will be displayed in a message box. If no <buttons> are specified, the user must click OK to acknowledge the message. If one or more <buttons> are specified and the /NMD switch is not used, then the <text> will be displayed in a message box that also includes each of the specified buttons. The user must select one of the buttons before the script will continue. If multiple <buttons> are specified and the /NMD switch has been used, then the script will proceed as if the user had selected the first specified <button>.

Script Argument	Action

MESSAGE *(continued)*	The following are the valid buttons: YES, NO, OK, CANCEL, ABORT, RETRY, IGNORE, HELP, YESALL, NOALL. Each specified <button> may optionally be followed by an equal sign then a <label> or the word ABORT. The word GOTO may optionally precede the <label>. If the user selects a <button> that is followed by the word ABORT, then the application will abort with error #7 (User Cancel). If the user selects a <button> that has a <label> in its definition, then the script will proceed at the specified <label>. If the selected <button> does not have a <label> or ABORT command, then the script will continue with the next line in the script. Note that the buttons have no intrinsic meaning. If the user selects the CANCEL or ABORT button, the script is not automatically cancelled unless the action of the button is ABORT or a GOTO that causes a jump to a label on the last line of the script. All of the button specifications, if any, must occur before <text>.
	If the message you want to display is too long to fit on a single line, you can break <text> into multiple strings and place the word NL (meaning new line) between the strings. For example, `MESSAGE "FIRST LINE" NL "SECOND LINE"` will display a two-line message.
	`SELECT FREESPACE FIRST`
	`SELECT IMAGE ALL`
	`RESTORE`

Script Argument	Action

MESSAGE *(continued)*	Example
	This is an example that asks the user if he would like to create a backup before deploying a new image. (The message text wraps in this example only to fit the page of the manual.)
	`SELECT DRIVE 1`
	`MESSAGE /YES /NO=GOTO Restore "Do`
	`you want to back up your system`
	`before deploying the new image?"`
	`SELECT PARTITION ALL`
	`SET IMAGE FILENAME "n:\backup.pqi"`
	`STORE WITH COMPRESSION HIGH`
	`:Restore`
	`DELETE ALL`
	`SET IMAGE FILENAME`
	`"n:\newimage.pqi"`

PROTECT PARTITION [DIAGNOSTIC | FIRST | LAST | UNKNOWN]	Protects the specified partition from being deleted by the DELETE ALL command. DIAGNOSTIC: First valid partition of an unknown type FIRST: First LAST: Last UNKNOWN: First valid partition of an unknown type
REBOOT	Reboots the computer; any commands following this command will not be executed. If the /NRB command line switch is specified, the computer will not actually reboot, but any commands after this command will not be executed.

Script Argument	Action
RESIZE EXTENDED {MAX | <number>}	Resize the extended partition, already on the disk, to the specified size in MB. When increasing the size of the partition, this is limited to the size of the partition plus the size of the free space around it. If the partition does not have any free space around it, then it will not be resized larger, even if there is other free space on the disk. When decreasing the size of the partition, the new size will be limited so that it must be as big or bigger than the partitions contained by the extended partition. MAX: Resizes to the maximum size possible. The right edge of the partition will be flush with end of drive or next partition, if possible. NUMBER: Resizes to value specified (in megabytes). The partition will be resized to the next smaller cylinder boundary, so it may not match the MB you specify exactly.

Script Argument	Action
RESIZE IMAGE {NO | PROPORTIONAL| <number> | MAX |MOST SPACE}	NO: Causes the last selected image to **not** be resized when the RESTORE command is encountered. Proportional: Proportionally resizes the last selected image when the RESTORE command is encountered. NUMBER: Resize the selected partition on the disk to the specified size in MB. When increasing the size of the partition, this is limited to the size of the partition plus the size of the free space around it. If the partition does not have any free space around it, then it will not be resized larger, even if there is other free space on the disk. When decreasing the size of the partition, the new size will be limited so that it must be as big or bigger than the data contained within the partition. MAX: Resize the selected partition on the disk to fill up all of the free space around it. If the partition does not have any free space around it, then it will not be resized, even if there is other free space on the disk. MOST SPACE: When the RESTORE command is encountered, resizes the partition with the most free space. 1) Ignores “Resize Image Proportional” and treats it as “Resize Image No.” 2) Ignores “Resize Image Max” and treats it as “Resize Image No.” 3) Considers "Resize Image <number>” as valid, causing the partition to be excluded from consideration when calculating the MOST SPACE.
RESTORE	Downloads all selected images into the free space and resizes them according to the RESIZE commands, if any.

Script Argument	Action
SECTOR CHECK {ON | OFF}	ON: Enables Bad Sector Checking for all restore operations following the command. OFF: Disables bad sector checking for all restore operations following the command. Bad sector checking is turned off by default.
SELECT DRIVE <number>	Selects the drive of the number specified. For example, SELECT DRIVE 1 selects the first hard drive in the system. After this command, all other commands refer to drive 1 until another SELECT DRIVE command is given. You can use the /INFO command line switch to display drive numbers.
SELECT FREESPACE {FIRST | LAST | NEXT | LARGEST}	Selects the specified free space on the currently selected drive. When selecting the largest free space, it does not matter whether the free space is inside or outside of the extended partition.
SELECT IMAGE {<number> | all}	Selects the specified partition in the image file. For example, SELECT IMAGE 3 or SELECT IMAGE ALL.

Script Argument	Action
SELECT PARTITION {<driveletter> | <number> | "<volumelabel>" | ALL | FIRST | NEXT | PREV | LAST}	Selects the specified partition. <driveletter>: Selects the partition assigned the specified drive letter by DOS or NT. Since DOS only assigns drive letters to visible FAT (and depending upon the version of DOS, possibly FAT32) partitions, only these types of partitions can be selected using this command. NT will assign drive letters to FAT/FAT32 and NTFS partitions. <number>: Selects the specified partition by number on the currently selected drive. Partitions are numbered in the order of their starting sector number on the disk, excluding any extended partitions or free spaces. You can use the /INFO command line switch to display partition numbers. "<volumelabel>": Selects the first partition on the currently selected drive with the specified volume label. For example, SELECT PARTITION "DATA" selects the first partition labeled "DATA." If more than one partition has this label (including FAT/FAT32,and NTFS partitions), only the first one is selected. You can use the /INFO command line switch to display volume labels. ALL: Selects all the partitions on the currently selected drive. FIRST: Selects the first partition. NEXT: Selects the partition immediately following the last selected partition, regardless of the syntax used to select that partition. PREV: Selects the partition immediately before the last selected partition, regardless of the syntax used to select that partition. LAST: Selects the last partition.

Script Argument	Action
SET ACTIVE	Sets the last partition selected as the active partition (meaning it will be the boot partition). The last partition selected must have been a primary partition.
SET DESCRIPTION "<*description*>"	Using this command before a STORE command will add <description> as a comment inside of the PQI file that is created. <description> should be enclosed by quotes.
SET IMAGE FILENAME "<*file*>"	Specifies an image file. Any previously specified image file will no longer be used with the script commands that follow. The newly specified file does not need to exist if you are doing a STORE command to create a new image file, but must exist if you will be reading from the file using a command such as RESTORE. The image filename should be specified in the same way that it would be specified for the /IMG command line switch. The name of the file should be placed in quotes.
SET PASSWORD "<*password*>"	Specifies the password to use when creating or restoring an image. The specified password replaces any previously specified password, including any password specified using the /PWD command line switch. This command is generally used along with the SET IMAGE FILENAME command to specify a new password to be used with the new image file. The password should be placed in quotes.

Script Argument	Action
STORE [WITH COMPRESSION HIGH | WITH COMPRESSION LOW | WITH COMPRESSION OFF]	Stores selected partitions with the indicated compression level. If the /CMP switch is specified, its setting will supersede the values specified by this command. No parameter specified: Stores selected partitions using the compression level specified by the command line switch /CMP. If the /CMP switch is not specified, then stores the image without compression. WITH COMPRESSION HIGH: Stores selected partitions with high compression. WITH COMPRESSION LOW: Stores selected partitions with low compression. WITH COMPRESSION OFF: Stores selected partitions with no compression.
UNHIDE	Unhides the currently selected partition.
VERIFY IMAGE	Validates that the image file, specified by the /IMG command line parameter or the SET IMAGE FILENAME script argument, is not corrupt.

Conditional Script Flow

The three commands LABEL, GOTO, and IF can be used for conditional script flow. GOTO is used in conjunction with the LABEL command.

LABEL

A LABEL command begins with a colon (:) in the first character of a line and is followed by the label name. A label can be up to 20 characters in length. All 20 characters are considered significant. Spaces are not permitted.

Example:

```
:THISISALABEL
```

GOTO

The GOTO command unconditionally transfers control to the label specified. The label may be defined anywhere in the script file. If the label is not found in the script file an error is generated and script file processing is terminated.

Example:

```
GOTO END
…
:END
```

IF

The IF command conditionally transfers control to the label specified. If the conditional test is true the GOTO command is executed. If it is false, control passes to the next line in the script. The IF command syntax is:

```
IF <operand> <comparison-operator> <operand> GOTO <label>
```

where: <operand> is a query keyword, environment-variable or constant.

You can also use the following syntax:

```
IF <query> GOTO <label>
```

where: <query> returns a Boolean value.

Example:

```
IF IsFAT32 GOTO CONVERT
…
:CONVERT
```

Strings

Strings can be used in comparisons with OS environment variables in IF commands. Strings are limited to 80 characters and must be delimited by double quotes. The syntax is:

```
"xxxx" where xxxx are the string characters
```

Example:

```
IF %MyENV% = "MyString" GOTO xxxxx
```

Comparison Operators

IF statements use the following comparison operators.

Description	Symbol
Equality	=
Inequality	<>
Less than	<
Greater than	>
Less than or equal to	<=
Greater than or equal to	>=

Queries

Queries are predefined for use in IF expressions. Values for queries are computed based on the currently selected disk and partition. Queries relating to volumes are not defined.

Disk Queries

Query Name	Used to	Returns
GetTotalDisks	Get the total number of physical disks on the selected disk	Number
GetTotalPartitions	Get the total number of partitions on the selected disk	Number
GetTotalUnallocatedSpaces	Get the total number of unallocated spaces on the selected disk	Number
GetDiskSize	Get the size of the currently selected disk in MB.	Number
GetAllocatedSize GetAllocatedPercent	Get the size of disk allocated to partitions in MB or as a	Number
GetUnallocatedSize GetUnallocatedPercent	Get the size of unallocated space on the currently selected disk in MB or as a percentage	Number

Partition Queries

Query Name	Used to	Returns
GetSelectedPartitionSize	Get the size of the partition in MB	Number
GetUsedAmount GetUsedPercent	Get the size of used space on a partition in MB or as a percentage	Number
GetUnusedAmount GetUnusedPercent	Get the size of unused space on a partition in MB or as a percentage	Number
IsFAT IsFAT32 IsNTFS IsHPFS IsLinuxExt2 IsLinuxExt3 IsLinuxSwap	Determine if the file system of the current partition is the type specified	Boolean

Query Name	Used to	Returns
IsActive IsHidden	Determine if the status of the current partition is active or hidden	Boolean
IsPrimary IsLogical	Determine if the class of the current partition is primary or logical	Boolean

OS Environment Variables

Operating system environment variables can be used in script commands. The value of the environment variable can be a number (such as for %NewSize%) or a string (such as for %MyPQIfile%. You can use environment variables for the following commands:

- Select Drive
- Select Partition
- Select Image
- Resize Image
- Resize Partition
- Set Image Filename

For example, in place of

```
Resize Partition 500
```

you can specify the following:

```
Resize Partition %NewSize%
```

Examples

Below are some examples of OS environment variables, including how the conditional script flow commands can be used.

Example 1 :

```
Resize Partition %NEWSIZE%
```

Example 2:

```
IF  GetSelectedPartitionSize < 2.1 GOTO SMALLER…
…
:SMALLER
```

Example 3:

```
IF %MYENVVAR% =  "TODAY" GOTO :MYLABEL
…
```

```
:MYLABEL
```

Example 4:

```
IF GetTotalPartitions < 2 GOTO SIMPLE
…
:SIMPLE
```

Script File Examples

Script files are just text files, so they can be created and edited using any application that you prefer. Symantec recommends using an extension of .PQS for these files so that they can be easily distinguished from other types of text files.

Scenario 1

Objective: To store all the partitions on drive 2 to the file, E:\IMAGES\DRIVE2.PQI:

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVE2.PQI
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 2
SELECT PARTITION ALL
STORE
```

Scenario 2

Objective: To store partition 3 from drive 1, and partitions 1, 4 and 5 from drive 2 to the file, E:\IMAGES\DRIVES.PQI and have the image file split up into multiple files each 650 MB so each can be transferred to a CD:

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVES.PQI /MFS=650000000
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 1
SELECT PARTITION 3
STORE
SELECT DRIVE 2
SELECT PARTITION 1
SELECT PARTITION 4
SELECT PARTITION 5
STORE
```

Scenario 3

Objective: To restore all the images in the file, /IMAGES/DRIVES.PQI, to the first free space on drive 2:

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=/IMAGES/DRIVES.PQI
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 2
SELECT FREESPACE FIRST
SELECT IMAGE ALL
RESTORE
```

Scenario 4

Objective: To restore the first two images in the file, E:\IMAGES\DRIVES.PQI, to the largest free space on drive 1 and resize them both proportionally

```
PQIDPLYD /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVES.PQI
```

The SCRIPT.PQS file contents:

```
SELECT DRIVE 1
SELECT FREESPACE LARGEST
SELECT IMAGE 1
RESIZE IMAGE PROPORTIONAL
SELECT IMAGE 2
RESIZE IMAGE PROPORTIONAL
RESTORE
```

Scenario 5

Objective: To restore the first two images in the file, E:\IMAGES\DRIVES.PQI, to the last free space on drive 1 and resize them to 500 MB each (the image was encrypted using the password 12345678)

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVES.PQI /PWD=12345678
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 1
SELECT FREESPACE LAST
SELECT IMAGE 1
RESIZE IMAGE 500
SELECT IMAGE 2
RESIZE IMAGE 500
RESTORE
```

Scenario 6

Objective: To delete all existing partitions on drive 2 and then restore all the images in the file, E:\IMAGES\DRIVES.PQI, to the free space (the image was encrypted using the password HELLO)

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVES.PQI /PWD=HELLO
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 2
DELETE ALL
SELECT FREESPACE FIRST
SELECT IMAGE ALL
```

```
RESTORE
```

Scenario 7

Objective: To restore the first four images in the file, E:\IMAGES\DRIVES.PQI, to the last free space on drive 1, and then resize the first image to 500 MB, keep the second the same size, and resize the third and fourth images to proportionally take up the remaining free space

```
PQIDEPLOY /CMD=SCRIPT.PQS /IMG=E:\IMAGES\DRIVES.PQI
```

SCRIPT.PQS file contents:

```
SELECT DRIVE 1
SELECT FREESPACE LAST
SELECT IMAGE 1
RESIZE IMAGE 500
SELECT IMAGE 2
RESIZE IMAGE NO
SELECT IMAGE 3
RESIZE IMAGE PROPORTIONAL
SELECT IMAGE 4
RESIZE IMAGE PROPORTIONAL
RESTORE
```

RTC Files

A runtime configuration file (RTC) is supplied for each version of PQIDeploy. The RTC file has the same base name as the executable, but the .EXE extension is replaced with a .RTC extension. RTC files are commonly used to embed the PQI file password. Doing so hides the password from customers and protects the PQI file from restoration, except by those with access to the proper PQIDeploy RTC file.

PQIDeploy has the ability to “lock” itself to run only on computers with a specific BIOS or DMI string. BIOS information is specified in the RTC file. When locking is enabled, the system BIOS or DMI is checked before PQIDeploy is allowed to execute. If the lock test fails, a customizable error message is displayed indicating the failure. BIOS locking is supported in all versions of PQIDeploy.

The PQIDeploy RTC file may also include an expiration date, after which PQIDeploy will not work. If there is an expiration date, it will be displayed each time you run the program to make you aware of when the program will expire.

You must contact Symantec Professional Services for RTC file customization.

PQIDeploy Application Exit Codes

The following list shows the possible exit codes that may be returned by PQIDeploy:

Code	Description
0	No errors
1	No errors, but a reboot is required before accessing modified partitions (corresponds to error code 33)
2	Insufficient memory (corresponds to error codes 3, 8, or 9)
3	Bad argument (corresponds to error code 4)
5	Wrong user (do not have administrator privileges)
11	Image corruption (corresponds to error codes 1800-1819, except 1810)
12	Unable to open image (corresponds to error codes 1810, 1831, 1850, 1851, 1852)
13	Scripting errors (corresponds to error codes 1820-1839, except 1831)
255	All other errors

PQIDeploy uses the same error codes as those found in ImageCenter or Drive Image. If you encounter error messages while using PQIDeploy, go to *service.symantec.com*, and choose Master Error List to display information about the error and the solution.

APPENDIX A

PQDisk Error Codes

Overview

There are two types of error codes returned by PQDisk:

- MS-DOS command shell ERRORLEVEL codes
- Internal Symantec error value codes

The MS-DOS command shell ERRORLEVEL must be an 8-bit number and is limited to 0-255. This value can be tested in a batch command file to determine program execution status.

If the ERRORLEVEL value indicates an execution error, check the full Symantec internal error value saved in the error file. The error numbers and their short description will also appear in the log file.

ERRORLEVEL Codes

The PQDisk ERRORLEVEL return error values:

Number	Description
0	Operation was successful
2	Out of memory or memory conflict
3	Invalid command-line argument used
4	Operation was cancelled
5	The partition table has one or more errors
6	The selected file system has one or more errors
13	Scripting errors
255	Miscellaneous, catch-all for other errors (check the error and log files)

Internal Symantec Errors

PQDisk internal error messages and some possible solutions are listed in error number order. The errors are also grouped in number ranges by error category. Errors with possible solutions are listed in bold. Other errors are listed in a normal font for completeness.

Miscellaneous Errors (1–39)

#1 Operation not supported

#2 This operation is illegal

#3 Not enough memory

This error can occur when you are resizing, moving, or copying a FAT32 partition. For more information about memory requirements, see "System Requirements". The DOS PQDisk executable requires a minimum of 585 KB of memory in the first 640 KB of the computer's address space (conventional memory) and 8 MB of total memory.

#4 Bad argument/parameter

#7 Operation cancelled by user

#8 Could not allocate/deallocate DOS real mode memory

The DOS PQDisk executable running under DOS, Windows 3.x, Windows 95, and Windows 98 requires some memory in the first 1 MB of the computer's address space (PQDisk uses a DOS extender). If not enough memory is available, PQDisk cannot access the hard disk.

#9 Not enough memory to complete the operation.

#10 At end of file

#20-22 PQ library errors

#23-24 Unsupported version of operating system

#25 The resource or service may not be accessed

#27 Cannot lock drive

Under multitasking operating systems such as Windows 9X, PQDisk must lock a partition before it can safely modify it. If the hard disk contains files that are in use by another process, PQDisk cannot lock the partition.

#28 Cannot unlock drive

#29 Cannot lock a locked drive

Verify that the partitions you are attempting to modify are not on a locked hard disk.

#30 Cannot unlock a locked drive

#33 The machine must be rebooted now (corresponds to ERRORLEVEL 1 above)

#34 The time has expired on this evaluation version

Symantec occasionally releases beta versions and evaluation versions of PQDisk. Both versions are not as safe as release versions; therefore, Symantec builds an expiration date into each version. After a predetermined test period, the beta or evaluation version no longer functions.

#35 Unknown file system

#36 DOS protected-mode interrupt error

#37 File is open on partition

#39 Error notifying the operating system of changes

Disk Access Errors (40–56)

Errors in the 40–56 number range indicate that accessing your disk is not possible, and often result from hardware problems. Some problems may have simple solutions; for others, the only solution may be replacing the hard disk. When possible, PQDisk detects major errors before any changes have been made so you can back up your data before replacing the hard disk.

#40 A hard disk is present but can't be accessed
#41 Disk is write protected
#42 Disk changed since last operation
#44 Drive not ready
#45 CRC error in data

When PQDisk or any other program reads information off of a hard disk, it checks the CRC (cyclic redundancy check) information contained in each sector. If it performs a CRC test and the result is different from the value stored on that sector, there is a CRC error . This usually means one of two things.

- The file being read has become corrupted by some other means.
- A sector used in the file's storage has become bad and corrupted that part of the stored file.

The solution is to do a surface test to make sure any bad sectors are marked as bad, then reinstall the software involved to ensure that files on the system are not corrupted.

#46 Seek error
#47 Unknown media type
#48 Sector not found

This error can be reported when a given sector cannot be read or written to. There are many possible causes. If you are encountering this error, make sure that your BIOS supports the operating system and hard disk on the system. Also run a thorough ScanDisk on the drive to prevent data from being written to bad sectors.

#49 Write fault
#50 Read fault

(The following information applies to errors 49 and 50.)

PQDisk is unable to write to/read from a specific sector on the hard disk. Possible causes include:

- If your PC beeps or displays a black box in the middle of the screen, virus protection is enabled in your computer's BIOS. Disable virus or boot sector protection in the BIOS.
- A virus protection application (which may be a TSR or DLL program) is in use. Disable the application before using PQDisk.
- There is a bad sector on the hard disk (this is usually the case only with older hard disks). Run ScanDisk on the hard disk to perform a surface scan to verify the existence of bad sectors. If your drive has bad sectors, we recommend you replace it.

- You have set up disk mirroring with PC-Tools. Disable the disk mirroring option.

#51 General hard disk failure
#52 DMA error
#53 Lock violation
#54 Network device fault
#55 Specified drive is busy
#56 Specified sector not in valid read or write range
#57 Cylinder number is too large for BIOS
#58 Unable to write to the boot sector

Miscellaneous Errors

#70 Changes to the drive cannot be made under Windows

You cannot run PQDisk under Windows. You must run PQDisk from a DOS prompt.

#71 This application will not run under DesqView
#72 This application will not run within a DOS Shell
#73-74 This application will not run under OS/2 DOS
#78-79 Read only Media
#83 Error trying to create a new process.
#84 Drive set to read-only for Symantec programs
#86 Unable to determine the BIOS drive order
#87 Unable to initialize Symantec engine
#88 Unable to find correct drive
#89 EZ-Drive has been detected on this drive but is not running
#90 EZ-Drive has been detected on this drive but is corrupt
#91 DiskManager has been detected on this drive but is not running
#92 DiskManager has been detected on this drive but is corrupt
#98 Windows 2000 was last shut down in hibernation mode.
#99 Windows Me was last shut down in hibernation mode.

Partition Table Errors (100–199)

Errors in the 100–199 number range are partition table errors.

#100 Partition table is bad

The master boot record (MBR) can contain, at most, one extended partition, and each extended partition boot record (EPBR) can contain, at most, one link to another EPBR. This error occurs when a partition table violates the foregoing rule. It can also occur if you have more than one active partition. Since any modifications PQDisk makes may decrease the amount of data that is recoverable from the hard disk, PQDisk does not recognize any of the hard disk's partitions. If you must create new, error-free partition tables to resolve your problem, see "Resolving Partition Table Errors" on page 199.

#102 Boot sector information not correct
#103 Could not create PQ sector (bad sector)
#104 No sectors in partition

No partition should contain zero sectors. Delete the partition before using PQDisk.

#105 Partition starts on wrong boundary

#106 Partition doesn't start with sector one

(This description applies for errors 105, 106, and 108.) The hard-disk partition table contains erroneous values. PQDisk expects partitions to begin and end on the correct cylinder boundaries. If they do not, the disk may be partially corrupted. In this circumstance, if PQDisk were to make any modifications it might cause the loss of data. Therefore, PQDisk refuses to recognize any of the hard disk's partitions. To resolve this problem, see the instructions in "Resolving Partition Table Errors" on page 199.

#107 Partition begins after end of disk

This error can occur if a partition erroneously extends beyond the physical end of the hard disk. This may happen if the hard disk has been used on a different computer or with a different hard-disk controller or if BIOS settings have been changed. Be advised that the physical geometry of the hard disk may differ from the logical geometry assigned to the hard disk by the operating system.

#108 Partition doesn't end at end of cylinder

See error #105.

#109 Partition ends after end of disk

See error #107.

#110 Partition table number of sectors is inconsistent

The hard-disk partition table contains two inconsistent descriptions of the number of sectors on the hard disk. This error is serious if both DOS and another operating system use the hard disk. Because DOS uses one description and other operating systems may use the other, data loss is likely once the partition is almost full. To resolve this error, see the instructions in "Resolving Partition Table Errors".

#111 The order of entries in the EPBR is not correct.

#112 Logical partition ends outside Extended

(This description applies for errors 111 and 112.) An extended partition boot record (EPBR) is a sector on the hard disk that contains a partition table. The EPBR partition table is special because it generally only has two valid entries: one for the logical partition and one that is a pointer to the next EPBR. The standard is for the logical partition's entry to be the first entry in the table and the second entry is the pointer to the next EPBR. The third and fourth entries are not used. For some utilities, such as IBM's Boot Manager, the order of these entries is important because the utility expects the first entry to be the logical and the second entry to be the pointer to the next EPBR. If PQDisk detects that the EPBR entries are out of order, you will be prompted to fix the error. If you choose to fix the error, PQDisk will reorder the EPBR entries for you automatically.

#113 Partitions overlap

The hard-disk partition table contains erroneous values. If data partitions overlap, writing to one may destroy data in another. This error is sometimes the result of an OS/2 FDISK bug. If free space exists within the extended partition, OS/2's FDISK program allows a primary partition to be created that overlaps the extended partition. A logical partition is subsequently created in the space occupied by the overlapping primary partition. If a primary partition overlaps the end of the extended partition but does not overlap any logical partitions within the extended partition, the problem can be remedied by patching the partition table. **Only qualified individuals should attempt this repair! An incorrect patch could destroy all data on the hard disk!** In most instances, you should resolve the problem as explained in “Resolving Partition Table Errors”.

#114 Start of Logical vs. EPBR is not correct

#115 End of Logical vs. end of EPBR is not correct

#116 Partition table Begin and Start inconsistent

The hard-disk partition table contains two inconsistent descriptions of the partition’s starting sector. This error can occur if the operating system reports a hard-disk geometry that is different than the geometry in use when the partition table was written. Possible causes include: (1) different operating systems (for example, DOS and Windows) report different hard-disk geometries, (2) you boot from a diskette that loads a different driver than is loaded when you boot from the hard disk, (3) upgrading the operating system causes a different driver to be used, (4) the hard disk or controller has been changed, (5) the BIOS has been upgraded, (6) the BIOS LBA setting has been changed, or (7) there is a partition table virus present on the hard disk. In most instances, you should resolve the problem as explained in “Resolving Partition Table Errors”. You can also use a virus scanning program to remove any partition table virus. Data loss is possible if the number of heads or sectors per track has changed since you first created your partitions.

#118 Two partitions have the same serial number

#119 A drive has been formatted since starting Norton PartitionMagic

#120 The logical drive chain is incompatible

This error occurs under some operating systems when logical partitions are not chained together in the expected order. DOS, OS/2, Windows 95, Windows 98, Windows NT, and Windows 2000 require that logical partitions be chained together in ascending order. Some other operating systems do not require this. For example, some versions of the Linux FDISK utility chain logical partitions together in the order they are created. This error message identifies a very dangerous situation; using the DOS FDISK in this situation can cause loss of one or more partitions. For solutions to this problem, see the instructions in “Resolving Partition Table Errors”. If you decide to back up your data and recreate your partitions, you may have to use the same partitioning program that you used to create the partitions in order to delete them.

#121 The first sector of the drive cannot be read

The first sector of the hard disk (cylinder 0, head 0, sector 1) contains the master boot record (MBR) and the primary partition table. PQDisk cannot make changes to this hard disk because an error occurred when it read the first sector. See error #50 for information on resolving this error.

#122 **A bad sector was found in the current or new partition area**

The partition cannot be moved safely because there is a bad sector in the new or current partition area. When you see this error message, the move operation is aborted before any corruption can occur. Try moving the partition to a different place. If your hard disk has bad sectors, we recommend that you replace the hard disk.

#123 **The free space is too small to copy the partition to it**

#124 The copy failed because the copy could not be created

#125 Some operating systems will not boot beyond 1024 cylinder boundary

#126 Unable to copy the extended since one already exists

#127 There is a drive letter conflict between PM and the OS

#128 The operating system is reporting a non-standard disk geometry

#129 Partition conversion is not available for this partition.

Boot Sector Errors (130–136)

#130 **Boot sector does not contain 0xAA55 ending signature**

#131 Boot sector does not have valid jump code

#132 Boot sector does not have valid OEM name

#133 Boot sector does not have valid file system type

#134 Boot sector does not have valid FAT/FAT32 signature

#135 Boot sector does not have valid Ext2 signature

#136 Boot sector does not have valid Linux Swap label

#182 Unable to determine a default working directory

#183 Unable to identify the Windows partition

#184 Unable to identify the Windows NT system partition

#185 A system (OS) service or function failed

#186 Unable to prepare machine for reboot

File I/O Errors (300–399)

#300 **A file with the specified name already exists**

#301 The file could not be opened

#302 The file could not be created

#303 The file could not be closed

#304 The file could not be read from

#305 Less bytes were read from the file than desired.

#306 The file could not be written to successfully

#307 Less bytes were written to the file than desired

#308 Could not seek to the file location specified

#309 Unable to get the current read/write position of the file handle

#310 Unable to get the size of the file

#311 Unable to set the file attributes

#312 Unable to get the file attributes

#313 Unable to rename the file

#314 Directories containing files or subdirectories cannot be deleted.

#315 Unable to get the file date and time

#316 Unable to set the file date and time

#317 The specified name is not a directory

#318 The specified name is not a valid file

#320 No drive letter in the path specified

#321 The specified file system has no file I/O functions
#330 File size greater than 32 bit value (4294967296)
#335 Could not uncompress the specified file

Check Errors (500–599)

Check errors occur when PQDisk checks the integrity of a partition. For general information about resolving these errors, see "Resolving Check Errors" on page 200.

#500 Subdirectory is corrupted

This error message reveals the name of the corrupted subdirectory. Back up the contents of that directory and its subdirectories. You can then delete the corrupted subdirectory.

#501 Cross-linked files were found

Multiple files claim the same clusters. PQDisk can fix this error when it occurs on an NTFS partition. For more information, see "Set Partition Table Autofix" on page 122. PQDisk lets you fix this error by: (1) copying the shared clusters to each affected file, (2) deleting all affected files, or (3) keeping one file and deleting the other affected files.

#502 Name already exists in directory
#503 Path was not found
#504 File not found
#505 Path was invalid
#506 Not enough free space on partition to shrink

Some free space (which is dependent on the hard disk's current contents) is required to resize a partition smaller. Delete unneeded and duplicate files in the partition and then attempt the operation again.

#507 File system not yet supported
#508 As specified, the operation does not change the partition

You have entered a value that is the same as or (when rounded to the required cylinder boundary) rounds to the same as the partition's present value. Enter a larger change.

#509 A bad sector was detected in the current or new FS area

To perform the resize operation that you requested, PQDisk attempted to expand the file system area. However, the program found a bad sector in the new area. Try moving the partition before you resize it. No corruption occurs when you encounter this error.

#510 The version of the file system is not supported

An updated version of PQDisk is required to operate on this new version of the file system. Visit *www.symantec.com* for information about updated versions of PQDisk.

#511 Not enough free space on partition to resize
#512 Not enough free space on partition to change cluster size
#513 Not enough free space on partition to make changes
#514 Unknown error formatting volume

Scripting Errors (950–999)

#948 Unable to get information for the specified volume segment.

#949 Unable to get information for the specified volume set.

#950 Unable to detect any disk drives

No partitionable hard disks were found on your computer. Diskette drives and many removable media drives do not support partitioning. PQDisk cannot perform operations on disks in such drives.

#951 An invalid value was entered

The value entered is outside the range or (when rounded to the required cylinder boundary) rounds to a value that is outside the range for the operation specified. Check the displayed range and reenter the value.

#952 Value entered is the same as the current value

See error #508.

#954 Partition is already of the type specified for convert

#955 Invalid parameter entered. (Parameters must start with '/' or '-')

#956 Unknown parameter entered.

#957 Could not open log file.

#958 Could not open script or command file.

#959 No Run File was found, program will not run

#960 Partition label was not unique

#961 Partition label specified was not found

#962 Invalid line in script

#963 Selected operation is currently invalid

Not all PQDisk operations can be performed on all partitions. For example, you cannot convert an HPFS partition to NTFS, and you cannot create a partition if there is not enough unallocated space on the hard disk. Under Windows, options that are not available either do not appear on the menus or they appear dimmed. However, if you are running scripts with the DOS version, there are no menus so you cannot see which operations are available. Refer to the relevant information in this user guide for restrictions that explain why an operation is not available.

#964 Tried to process unknown command

#965 Unable to select drive or partition with drive letter

#966 Could not open error file.

#967 Could not perform operation to the value specified

This error occurs only in the enterprise version of PQDisk from a running script. If the value specified on a resize or move operation is not between the minimum and maximum possible, script execution stops and this error displays.

#968 Incorrect Volume Label entered, Deletion not performed

To delete a partition, PQDisk requires you to enter that partition's volume label. If the volume label you enter does not match the volume label of the partition you want to delete, this error appears.

#969 Incorrect Volume Label entered, Unable to proceed.

To format an existing partition, PQDisk requires you to enter that partition's volume label. If the volume label you enter does not match the volume label of the partition you are attempting to format, this error appears.

#970 Invalid Bad Sector Check value specified

If the script command SET DEFAULT BAD SECTOR TEST STATE is not followed by either ON or OFF, this error appears.

#971 The label entered was too long

When you enter a volume label, the process that checks the validity of the label displays this message if the label is too long. The label must be no longer than 11 characters for FAT partitions, 16 for Ext2 partitions and 255 for .NTFS partitions.

#972 Invalid characters in the label

When you enter a volume label, the process that checks the validity of the label displays this message if the label has characters that are invalid. Invalid characters include the following: [* ? : < >|+ = ; \ / " ,].

#973 Volume Label cannot have leading spaces

When you enter a volume label, the process that checks the validity of the label displays this message if you enter a label in which a space or spaces are the leading characters.

#974 Root size specified was not in the valid range

If you use the Create, Format, or Resize Root operations, and the number of root entries specified is not within the acceptable range for that partition, this error appears. Generally, the valid range is from 64 to 1,024.

#975 The cluster size specified was invalid for this partition

Many commands have a cluster size option. If a script command specifies an invalid cluster size (for the type and size of the partition), this error appears.

#976 Cannot create the file system specified in the current space

When you use the Create or Format commands, you must also choose a file system type. If the file system or partition type you specified cannot be created in the space available, this error appears.

#977 Partition selected is invalid

This error message displays only in the enterprise version of PQDisk from a running script. If the partition selected from the Select Partition command is not a valid partition, this error appears.

#978 Unable to set to proper partition after last operation. Script halted.

After each operation, PQDisk ensures that the right partition is still selected. If PQDisk is not able to select the proper partition, it ends script processing and displays this error.

#979 Unable to perform requested operation, disk full.

#980 Invalid folder name.

#981 Attempt to copy used partition to used partition

#982 Free Space is not big enough to hold partition information

#983 Too many errors found, process halted.

#984 Invalid path specified

#985 File not found at specified location, or the path is not accessible.

#986 Unable to get information for the specified partition

PQDisk reports this error most commonly when MS-DOS-based terminate-and-stay-resident programs (TSRs), are running in the background. These TSR's will be located in the Config.sys or Autoexec.bat files.

One such TSR is the SUBST command. The SUBST can be used to associate a path with a drive letter. This creates a "virtual drive" that can be accessed as an additional local drive. The SUBST command is classified as a "dirty" or "deadly" TSR, and cannot be loading at the same time that PQDisk is loading. To solve the problem, you must remark out the Config.sys or Autoexec.bat line that is loading the SUBST command. This command can be reinstated after running PQDisk. There are other “dirty” or “deadly” TSR's that may cause a problem. If you are experiencing this error and are not using the SUBST command, find and remark out any of the following commands: Join, Append, or Assign.

This error can also be reported if your C: drive is compressed. If the C: drive is compressed, you will need to either uncompress the drive, or run PQDisk from the rescue diskettes. This error can also be caused by multiple partition table errors. If any of the above solutions do not apply, run PartitionInfo and send the report to Symantec technical support.

#987 Invalid Drive Read Only value specified.

#992 Cannot convert from FAT to NTFS - partition has no drive letter.

#993 Partition contains open files. Use the OS check utility.

#994 The partition must contain at least 2 root entries.

#995 Each partition must have at least one root entry.

HPFS Check Errors (1000-1099)

#1000 Extended Attributes invalid

#1001 Extended Attributes special file invalid

#1002 Found orphaned extended attribute

#1003 Error trying to initialize HPFS partition

#1004 The Fnode parent pointer is not correct

#1005 Fnode is not a directory, but should be!

#1006 Dirblock is a node, not a leaf

#1007 Dirblk sig or back ptrs not correct

#1008 First directory entry is incorrect

#1009 Sector in allocation sect not correct

#1010 Number of sectors not equal to file size

#1011 Allocation sect signature or self ptr not correct

#1012 SuperBlock not valid

#1013 SpareBlock not valid

#1014 Partition not formatted by OS/2 2.x

#1015 System sector not marked unavailable

#1016	Sector read error in dir structure
#1017	Sector read error in system structure
#1018	Root Fnode pointer invalid
#1019	Bitmap sectors not marked unavailable
#1020	OS/2 wasn't shut down properly
#1021	Hot fixes active -- run CHKDSK from OS/2
#1022	OS/2 marked partition as corrupt!
#1023	Too many hot fixes -- run CHKDSK from OS/2
#1024	Hot-fix sectors not marked unavailable
#1025	Reserved dirblks not marked unavailable
#1026	Error in code-page structure
#1027	Could not account for all sectors
#1028	Counted more sectors than exist on partition!
#1029	Error writing bitmap to disk
#1030	Allocation sector's parent pointer invalid
#1031	Could not move internal disk structures
#1032	Could not move the Directory Band
#1033	Could not shrink partition
#1034	List of bad sectors is corrupted
#1035	Too many bad sectors on partition
#1036	Storage block has wrong # of allocations
#1037	Directory entry is corrupt
#1038	Directory block is corrupt
#1039	Error reading into DirBlk cache
#1040	DirBlk sector must be multiple of 4
#1041	SuperBlk.ulSectsOnVol > Bs.bsHugeSects
#1042	Incorrect # of hotfix sects
#1043	Parent of dirblk is not fnode
#1044	Number of sectors in allocation cannot be 0
#1045	Stac volume detected -- delete Stac volume before converting!
#1046	Cannot allocate sector on disk
#1047	Invalid or unknown HFPS version
#1048	Partition is too small cannot format
#1049	Cannot format -- bad system area

Linux Ext2 and Ext3 Check Errors (1100-1233)

#1100	Miscellaneous ext2 library error
#1101	Ext2 functionality not implemented
#1102	Internal ext2 library error: incorrect structure magic number
#1103	Bad magic number in super-block
#1104	Attempt to write to file system opened read-only
#1105	Can't read Ext2 file system data
#1106	Can't write Ext2 file system data
#1107	Corrupt Ext2 group descriptor
#1108	File system has unexpected block size
#1109	Ext2 directory corrupted
#1110	No free space in the directory
#1111	Illegal inode number passed to ext2 library
#1112	Illegal block number passed to ext2 library

#1113	**Not enough space to build proposed file system**
#1114	**Illegal indirect block found**
#1115	**A block group is missing an inode table**
#1116	**The Ext2 superblock is corrupt**
#1117	**Too many symbolic links encountered.**
#1118	**The inode is from a bad block in the inode table**
#1120	**Could not allocate block in ext2 filesystem**
#1121	**Could not allocate inode in ext2 filesystem**
#1122	**Ext2 inode is not a directory**
#1123	**Too many references in table**
#1124	**File open read-only**
#1125	**Ext2 directory block not found**
#1126	**Ext2 directory already exists**
#1127	**Illegal bitmap blocks in ext2 file system**
#1200	**Ext2 file system is corrupt**
#1201	**Ext2 superblock contains illegal information**
#1202	**Adding UUID to filesystem**
#1203	**Root directory is corrupt**
#1204	**Root directory had dtime set**
#1205	**Reserved inode has bad mode**
#1206	**Inode is unused but does not a detection time set**
#1207	**Inode is in use but has a deletion time**
#1208	**Inode is a zero-length directory**
#1209	**Bad block in file system metadata**
#1210	**Inode has incorrect size field**
#1211	**Block has incorrect block count**
#1212	**Inode has an illegal block**
#1213	**Conflicting location of filesystem metadata**
#1216	**Directory is corrupted**
#1217	**Directory contained entry to a deleted file**
#1218	**Inode is corrupted**
#1219	**Directory disconnected from directory hierarchy**
#1220	**Zero-length inode disconnected from directory hierarchy**
#1221	**Directory disconnected from directory hierarchy**
#1222	**Inode has an incorrect reference count**
#1223	**Padding at the end of bitmap not set**
#1224	**Block not used, but marked in bitmap**
#1225	**Block used, but not marked used in bitmap**
#1226	**Inode not used, but marked in bitmap**
#1227	**Inode used, but not marked used in bitmap**
#1228	**Free inodes count for group wrong**
#1229	**Directory count for group wrong**
#1230	**Free inodes count wrong**
#1231	**Free blocks count for group wrong**
#1232	**Free blocks count wrong**
#1233	**Blocks claimed by multiple inodes**

NTFS Check Errors (1500–1699)

Errors 1500–1699 are NTFS-specific check errors, which can occur when PQDisk checks the integrity of a partition. PQDisk can fix certain errors when you perform the Check operation. For more information, see "Resolving Check Errors" on page 200. In this section, "attribute" does not mean read-only, hidden, system, etc. Rather, "attribute" means one of a file's data streams.

#1500 Bad boot signature bytes

#1501 Wrong version of NTFS

The partition was created using a version of the NTFS file format that PQDisk cannot work with.

#1502 Error writing volume bitmap

#1503 Bad NTFS cluster size

The NTFS cluster size must be 512, 1,024, 2,048, 4,096, 8,192, 16,384, 32,768, or 65,536 bytes.

#1504 Journal file too short

#1505 Error writing file table bitmap

#1506 Too many bad clusters

#1507 Bad file record signature

#1508 Bad directory buffer signature

#1509 Bad journal entry signature

#1510 Bad restart entry signature

#1511 Bad restart record format

#1512 Restart record mismatch

The two restart entries in the journal file are different. This may happen if Windows NT/2000 is not properly shut down. To fix this problem, restart Windows and shut it down using the Shut Down command.

#1513 Bad attribute position in file record

#1514 File attribute too short

#1515 File record last byte beyond end

#1516 Partition improperly dismounted

The partition dirty flag is set in a restart record in the journal file. This error may have been caused by a power failure or system crash while the Windows operating system was writing the partition. Reboot Windows and execute CHKDSK /F to repair the damage.

#1517 Required file attribute missing

#1518 File attribute doesn't fit

#1519 File attribute data doesn't fit

#1520 External attribute header mismatch

#1521 Attribute must be nonresident

#1522 Attribute must be resident

#1523 Cluster beyond end

#1524 Cluster number too large

#1525 Bad directory buffer size

#1526 Directory entry too long

#1527 Bad update sequence number

A buffer contains mismatched update sequence numbers. This error may have been caused by a power failure or system crash while the Windows NT/2000/XP operating system was writing to the partition. Reboot Windows NT/2000/XP Workstation and execute CHKDSK /F to repair the damage.

#1528 Name mismatch in directory entry

#1529 Information mismatch in directory entry

A file attribute stored in a file record is different from the attribute stored in its directory entry. If this error is in a system file (file 0–10), Windows NT/2000/XP CHKDSK does not fix it, but Windows NT/2000/XP rebuilds the root directory on the partition the next time the operating system is started.

#1530 Zero reference count

#1531 Too few clusters

#1532 Too many clusters

#1533 Allocated cluster list too long

#1534 Cluster not allocated

#1535 External attribute list wrong length

#1536 Nonresident attribute too short

#1537 File number greater than 4 billion

#1538 Can’t find contiguous space to move

The partition does not contain enough contiguous free space to hold the new copy of a file that must be contiguous. You normally encounter this error when you use the Resize option to resize a partition smaller.

#1539 File size mismatch

The size of a system file (file 0–15) recorded in its file record does not match either the size recorded in its directory entry in the root directory or the size of its data stream.

#1540 Run size mismatch

#1541 Bad file record size

#1542 Gap in multiple runs

#1543 Overlap in multiple runs

#1544 External attribute list in external attribute

An external file record has an external attribute list.

#1545 File attributes out of order

The attributes in a file must appear in order of increasing numeric type.

#1546 Attribute neither resident nor nonresident

The attribute resident flag has a value other than resident or nonresident.

#1547 Wrong run limits

A run has more clusters than the difference between its highest and lowest cluster.

#1548 File table has fewer than 16 entries

The file table must have at least 16 entries.

#1549 File table has more than 4 billion entries

The file table must have fewer than 4 billion entries.

#1550 Attribute instance number too large
#1551 Partition full
#1552 The requested FRS entry is unused (free)
#1553 The root directory does not contain an index
#1554 File has multiple short names in the same parent directory
#1600 Error in NTFS boot sector
#1601 Error in NTFS partition information
#1602 File table backup mismatch
#1603 System files not all in root File
#1604 File's parent does not contain the file

The file's parent directory does not contain a reference to the file, or a file's size, date, or time information does not match the file's parent directory information. This error can be fixed when you perform the Check operation. See "Check" on page 110. When you fix this error, PQDisk updates the file's parent directory information.

#1605 File's parent not directory
#1606 System file records not all marked used
#1607 File record marked free
#1608 File record marked used
#1609 Lost cluster(s)

The volume bitmap shows clusters as being used which are not used (no file claims them). This error can be fixed when you perform the Check operation. See "Check" on page 110. PQDisk lets you fix this error by either deleting the lost clusters or by saving them in a file in the root directory. The filename is FILEXXXX.PQE, where XXXX is a number between 0000 and 9999.

#1610 Volume bitmap overflow bits not set
#1611 Bad system file name
#1612 Directory flag not set
#1613 Directory flag set for non-directory
#1614 Bad boot record position
#1615 Bad center boot backup position
#1616 Partition full
#1617 No directory buffer
#1618 No directory buffer bitmap
#1619 Bad pointer in security descriptor
#1620 Backup boot record mismatch
#1621 Bad backup file table size
#1622 Boot record too short
#1623 Root not a directory
#1624 Bad root directory format
#1625 Index not sorted
#1626 Bad upcase table size
#1627 Upcase table incorrect
#1628 Attributes out of order
#1629 Bad attribute definition table format

#1630 Inconsistent sizes in attribute header

File size information is incorrect. This error can be fixed when you perform the Check operation. For more information, see “Check” on page 110. When you fix this error, PQDisk computes the correct file size information.

#1631 Volume bitmap wrong size
#1632 Volume bitmap not contiguous
#1633 Non-directory lacks data
#1634 Directory has non-directory data
#1635 Missing internal file name
#1636 Missing standard information
#1637 File table bitmap wrong size
#1638 Directory buffer bitmap wrong size
#1639 Unused directory buffer marked used
#1640 Used directory buffer marked unused
#1641 Directory buffer bitmap w/o buffer
#1642 Index sequence number mismatch
#1643 File name indexed flag not set
#1644 Bad system file sequence number

A system file has a bad sequence number. System files must have a sequence number from 1 to 15. A partition with this problem may pass Windows NT/2000/XP CHKDSK, but Windows does not mount the partition the next time the operating system is started.

#1645 Bad cluster file has nonzero size

#1646 Bad cluster attribute has wrong size

#1647 Error in root directory index

There is an error in the root directory’s index. Running Windows NT/2000/XP CHKDSK does not fix this problem, but Windows automatically rebuilds the root directory on the partition the next time it is started.

#1648 Log file contains less that 1Mbyte
#1649 Log file contains more that 4Mbytes
#1650 Partition too fragmented to restore
#1651 Index has nonzero length
#1652 Bad entry in external attribute list
#1653 System file compressed
#1654 File system is smaller than partition
#1655 FRS not in any directory
#1680 Extended index is present
#1681 Data is compressed or sparse
#1682 File attribute cannot be converted to FAT
#1683 File system is too small to convert to FAT-32
#1684 File system is too large to convert to FAT
#1685 Multiple long names or links
#1686 Multiple data streams
#1687 An object index is present
#1688 A text index is present

#1689 File has multiple versions. Only newest will be retained.
#1690 File has an ACL. Access will be unrestricted after conversion.
#1691 Free space of clusters less than clusters needed to convert to FAT
#1692 A file in directory is larger than 4GB
#1693 Partitions with 64k clusters cannot be converted to FAT
#1694 Bad clusters found at beginning of partition

FAT Check Errors (2000–2099)

Check errors occur when PQDisk checks the integrity of a partition. For general information about resolving these errors, see See “Resolving Check Errors” on page 200.

#2001 FAT copies are not identical

Run ScanDisk to fix this error. This problem may also be caused by a virus. Run a virus checker and remove the virus if possible.

#2002 There are invalid entries in the FAT

This error can generally be fixed by running a thorough ScanDisk on the partitions reporting the error.

#2003 File size does not match FAT allocation for file

Run ScanDisk or CHKDSK to fix this error.

#2004 An invalid cluster was found in a directory entry
#2005 One or more lost clusters are present

Run ScanDisk or CHKDSK to fix this error.

#2006 Too many clusters to allocate at once
#2007 Cluster size is > MAX_CLUST_BYTES
#2008 Not enough free clusters on disk
#2009 Couldn't find free dir entry in root
#2011 Entry cannot have both VOL and DIR bits set
#2012 Formatted FAT file system too big for partition

This error can occur when:

- The number of sectors in the partition is larger than 65,536, and the HugeSects field of the boot sector (“Big total number of sectors” in Norton’s DISKEDIT utility) shows that there are more sectors in the partition than the partition table shows.
- The number of sectors in the partition is less than 65,536, and the Sects field of the boot sector (“Total sectors on disk” in Norton’s DISKEDIT utility) shows that there are more sectors in the partition than the partition table shows.

This situation can result in data loss when the FAT file system tries to use space outside the partition that does not exist or that belongs to another partition. Since file data may exist outside the partition boundary, you cannot fix the problem by simply patching the boot sector.

To correct the error, back up all data on the partition, delete the partition, recreate the partition, and restore the data. Alternately, it has been reported that you can use Norton Disk Doctor to fix this problem.

#2013 A component of FAT geometry is bad

This error can occur when:

- The number of clusters on the hard disk is greater than the FAT limits allow. This can result from bad values in the boot sector for the number of sectors, FATs, root entries, reserved sectors, and sectors per cluster.
- The number of sectors in the FAT is not large enough to hold the number of clusters present on the hard disk. A qualified consultant may be able to fix the hard disk by performing simple patches. Alternately, you can back up the data on the partition, delete the partition, recreate the partition, and restore the files.

#2014 OS/2 is incompatible with the requested FAT size

#2015 The size of a cluster cannot change

#2016 An extended attribute allocation error was found. Run CHKDSK /f.

#2017 The specified EA index is outside the range of the EA file

#2018 The specified EA index is was allocated, but has been deleted

#2019 A file seek or read was attempted but the FAT is not loaded

#2020 The cluster alignment map is invalid

#2021 Data in the disk is not aligned as expected

#2022 Data was not cleared from clusters that will be marked bad

#2023 Data on the partition is preventing the operation

#2024 The OS/2 Extended Attribute file is corrupt

#2025 The requested partition parameters result in a root directory that is too large

#2026 The requested fat partition would have invalid parameters

#2027 Too many root entries in the FAT32 partition to convert it to FAT16

Long filenames may be causing this problem, since they use multiple entries per file. To fix this error, move some of the root directory entries into a subdirectory and defragment the disk.

#2100 **There are already 4 primary partitions**

#2101 There may not be enough space to create a primary below the 1024 cylinder boundary

#2102 Partition is too large to fit under 1024 cylinder boundary

#2103 Partition of specified size could not be created.

Operating System Errors (over 10,000)

Any number over 10,000 indicates an operating system error. To determine the number of the operating system error, subtract 10,000. See your operating system documentation for information about resolving the error.

Resolving Partition Table Errors

Partition table errors are errors in the 100 - 199 range. In most cases, you must resolve partition table errors by creating new, error-free partition tables. The general steps are as follows:

1. Ensure you have no viruses. See “Partition Tables and Viruses” on page 200.
2. Back up the data on the affected partitions.
3. Delete the partitions.
4. Recreate them.
5. Restore their contents.

You may need to use the FDISK program from a recent DOS version, as earlier versions may refuse to delete hidden partitions.

Partition Tables and Viruses

If partition changes made under one operating system are not reflected under another, and vice versa, a master boot record (MBR) virus may be present.

Use a virus check utility that can detect the latest viruses. If a virus is found, data loss is likely. Before removing the virus, boot each operating system and use the Check command to evaluate the integrity of the partition. Back up the files on any partition that passes the Check command. Then remove the virus and perform the Check command on the partitions again. Delete and recreate any partitions that fail the check. Finally, reinstall the operating systems and restore the backup files as necessary.

Resolving Check Errors

PQDisk checks the integrity of a partition thoroughly before making changes to it. The Check and Info commands perform the same checks and display error messages when they discover problems. These checks are similar to those made by an operating system’s CHKDSK, ScanDisk, or AUTOCHK utility.

PQDisk also checks a partition after modifying it. While data loss is possible, it is not typical. The problem is usually a minor file system error that CHKDSK /F /R (or ScanDisk, if you are using Windows 95/98) can correct without data loss. For more extensive errors, you may need to restore your files from a backup copy.

If you receive a Check error message on any partition, back up your hard disk and then run your operating system’s CHKDSK program on that partition (do not use the /F switch on the initial run). If you have MS-DOS 6.x, Windows 95, or Windows 98, run ScanDisk. CHKDSK and ScanDisk generally discover the same problems as PQDisk (except that the DOS CHKDSK program does not detect problems in Extended Attributes).

If CHKDSK or ScanDisk and the Check command detect the same errors, which is usually the case, run CHKDSK with the /F switch or run ScanDisk to fix the problems. Then run CHKDSK again without the /F switch to ensure that the partition is error free.

When CHKDSK reports no errors on the partition, run the Check command. If PQDisk still reports a problem, reformat the partition and restore your files from the backup copy.

Index