

Using BrightWorks

McAFEE

Copyright © 1995 by McAfee, Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the written permission of McAfee, Inc. 2710 Walsh Avenue, Santa Clara, California, 95051-0963.

McAfee is a registered trademark of McAfee, Inc. SiteMeter, BrightWorks, LAN Inventory, NetShield, Desktop Control Language and NetRemote are trademarks of McAfee, Inc. All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations.

Document Release BW.20

Table of Contents

Chapter 1 Introduction	9
About Your Manuals	9
Using BrightWorks.....	9
Using the Smart-Launch Modules	10
Reference Information	10
Manual Organization	11
About BrightWorks	12
BrightWorks' Features.....	12
Server Monitoring.....	14
Help Desk Automation	14
Environment	14
Chapter 2 Installation	17
Before Installation.....	17
Determining Version Numbers	18
BrightWorks Installation.....	18
Upgrade Install.....	24
Configuration Procedures	27
Loading NLMs	27
Chapter 3 Getting Started	29
The BrightWorks Console	29
Windows Terms.....	29
Accessing BrightWorks	30
Exiting BrightWorks.....	31
BrightWorks Menu Bar	31
BrightWorks Tool Bar	32
Using the Keyboard	34
BrightWorks' Help Facility	34
BWORKS.INI File.....	35
Printer Setup and Administration.....	36
Changing Print Settings	37
Distribution Tutorial	38
Creating and Compiling a Script	38

Creating a Scope	42
Scheduling a Package	43
Running the Update Program.....	46
Chapter 4 Software Distribution	47
About the Software Distribution Capability	47
Software Distribution Concepts	48
BrightWorks' Software Distribution Modules	50
BrightWorks Console/ Administrative Program	50
Update Program	51
Distribution Configuration Options	51
Running the Update Program.....	52
Chapter 5 Filesets	55
Introduction.....	55
Fileset Features	55
The Fileset Directory.....	56
Defining a Fileset Directory	56
Creating Filesets.....	58
Managing Filesets	61
Editing Filesets	61
Renaming Filesets.....	63
Copying Filesets	64
Deleting Filesets	64
Chapter 6 Scripts	65
Introduction.....	65
Script Features	65
Creating Scripts.....	66
Compiling Scripts	69
Compiling Scripts	69
Managing Scripts	71
Editing Scripts	71
Renaming Scripts.....	72
Copying Scripts	73
Deleting Scripts.....	73
Incorporating BrightWorks Scripts	74

Chapter 7 Software Distribution QuickScript Language	77
Introduction	77
Notes on Syntax.....	77
Script Functions.....	78
Function Set Overview	78
Type of Functions.....	78
DOS Functions	78
System File Functions	79
Windows System File Functions	80
Miscellaneous Functions	81
User-defined Variables	82
DOS Functions.....	82
ATTRIB [path] [filewild] [attribute]	83
COPY [path] [filewild] [path] {filewild}	84
DELETEDIR [path] [filename] {deleteopt}.....	84
DELETEFILE [path] [filewild].....	85
FINDFILE [path] [filewild] [strvar]	85
MDIR [path] [filename].....	86
RENAME [path] [filewild] [path] [filewild]	86
UPGRADEOS [upgopt].....	87
System File Functions.....	88
ADDDEVICE [strvalue1] [strvalue2] [addopt].....	89
ADDLINE [strvalue1] [strvalue2] [addopt]	89
ADDPATH [strvalue1] [strvalue2] [strvalue3] [addopt].....	90
CFGGETVALUE [strvalue] [intvar].....	91
CFGSETVALUE [strvalue] [intvalue].....	91
CFGRETRIEVESTSTRING [strvalue] [strvar].....	92
CFGSETSTRING [strvalue1] [strvalue2]	92
REPLACEKEY [strvalue1] [strvalue2] [strvalue3]	92
REPLACELINE [strvalue1] [strvalue2].....	93
REPLACELINEADD [strvalue1] [strvalue2] [addopt].....	93
SETSYSFILE [path] [filename].....	94
Windows System File Functions	95
ADDGROUP [strvalue]	95
ADDITEM [strvalue1] [strvalue2] [pathfile]	97
GETINIINT [pathfile] [strvalue1] [strvalue2] [intvar]	98
GETINISTR [pathfile] [strvalue1] [strvalue2] [strvar].....	98
SCHEDULEWIN [path] [filename] [text].....	99
WRITEINIINT [pathfile] [strvalue1] [strvalue2] [intvalue].....	99
WRITEINISTR [pathfile] [strvalue1] [strvalue2] [strvalue3]	99
Miscellaneous Functions.....	100
APPENDPATH [strvar] [strvalue]	100
ASSIGN [intvar] [intvalue].....	101
DEFINE [text] [defineopt]	101

EXIT [intvalue]	102
IF [intvalue1] [condoper] [intvalue2] ... {ELSE...} ENDIF	103
NUMTOSTR [strvar] [intvalue]	103
PAUSE [text]	104
REBOOT	104
SHELL [pathfile] {text} {shellopt}	104
STRCAT [strvar] [strvalue]	105
STRCOMPARE [strvar] [strvalue]	106
STRCOPY [strvar] [strvalue]	106
WRITELN [strvalue]	107
Rules and System Variables	107
Rules	107
System Variables	109
String Type Rules:	109
Integer Type Rules:	109
DOS Error Codes	110
Chapter 8 Scopes	113
Introduction	113
Creating Scopes	113
Scope Queries	115
Applying a Query to the Scope	116
Removing a Query from the Scope	117
Creating a Complex Query	117
Creating a New Query	119
Editing a Query	121
Deleting a Query	121
Managing Scopes	122
Editing Scopes	122
Renaming Scopes	123
Copying Scopes	124
Deleting Scopes	124
Chapter 9 Packages	125
Introduction	125
Package Features	125
Creating and Editing Packages	126
Creating Packages	126

Managing Packages	132
Prioritizing Packages	132
Activating/Deactivating Packages	134
Deleting Packages.....	134
The Package Timer	134
Setting the Package Timer Interval	135
 Chapter 10 Monitoring Software Distribution	 137
Introduction	137
The Software Distribution Log	137
Viewing Log History Details	137
 Appendix A Error Messages	 141
Script Editing Error Messages	141
Software Distribution Error Messages	142
Software Distribution Update Program Error Messages	144
Installation Troubleshooting	148
 Index	 151

Chapter 1 *Introduction*

Welcome to BrightWorks, the easy-to-use, comprehensive network management solution from the network utilities experts.

BrightWorks is a member of McAfee's family of intuitive network management tools, a group of network applications all designed to reduce the cost of network ownership.

About Your Manuals

Using BrightWorks

This manual provides an overview of your BrightWorks product including, installation procedures, console description, and procedures for using the software distribution capabilities.

To provide software metering and network inventory capabilities, BrightWorks fully integrates with McAfee's SiteMeter, LAN Inventory, Desktop Control Language, NetShield and NetTune products. The documentation for these products are included with your purchase of BrightWorks.

After completing the installation described in Chapter 2 of this manual, all of the McAfee management capabilities that integrate with BrightWorks are enabled. The procedures for metering network software from BrightWorks are identical to those from SiteMeter; and those for inventorying are the same for BrightWorks and LAN Inventory. The same holds true with the other McAfee products included with your purchase of BrightWorks. As such, you should refer to those manuals for complete instructions on using those components of the BrightWorks product.

The table below indicates which manual describes which capability:

BrightWorks Capability	Manual
Metering	Using SiteMeter
Inventory	Using LAN Inventory
Software Distribution	Using BrightWorks
Help Desk Support	Using LAN Support Center

Server Monitoring	Using NetTune
SD PowerScripts	Using PowerScript DCL

Using the Smart-Launch Modules

In addition to these capabilities, you can also access other McAfee management solutions from the BrightWorks console by choosing the appropriate tool bar button. Refer to “BrightWorks Tool Bar” in Chapter 3, “Getting Started” for instructions.

The table below indicates which tool bar button and manual should be used to access other McAfee products.

McAfee Product	Tool Bar Button	Manual
NetRemote	Remote	Using NetRemote
LAN Support Center	SupportCtr	Using LAN Support Center
DCL PowerScript	PowerScripts	Using PowerScript DCL
NetShield	NetShield	Using NetShield

Reference Information

Several reference chapters that pertain to BrightWorks are included in all three manuals. The following chart describes what information is available in each of the appendices.

Reference	Manual and Appendix
Software distribution error messages	<i>Using BrightWorks</i> Appendix A
Metering error messages	<i>Using SiteMeter</i> Appendix A
Inventory error messages	<i>Using LAN Inventory</i> Appendix B
Support Center error messages	<i>Using LAN Support Center</i> Appendix A
Equipment database lists	<i>Using LAN Inventory</i> Appendix A
Instructions on using and configuring Brequest	<i>Using LAN Inventory</i> Appendix C <i>Using SiteMeter</i> Appendix B
Lists of Btrieve status codes and their explanations	<i>Using SiteMeter</i> Appendix C <i>Using LAN Inventory</i> Appendix D
Instructions for the NMS Smart-	<i>Using LAN Inventory</i> Appendix E

Launch

NOTE: The procedures for generating reports on software distribution activity and inventory activity are the same. Therefore, refer to Chapter 9 in *Using LAN Inventory* for complete instructions on generating reports about software distribution activity on your network.

Manual Organization

Chapter	Description
Chapter 1: Introduction	Provides background information and environment requirements.
Chapter 2: BrightWorks Installation	Provides complete instructions for installing BrightWorks, including upgrade and NLM procedures.
Chapter 3: Getting Started	Describes the console, printer set-up and key Windows terms as well as provides a tutorial describing software distribution's features.
Chapter 4: Software Distribution	Provides an overview on the use of BrightWorks' software distribution capabilities. Also provides instructions on running and automating the distribution update program.
Chapter 5: Filesets	Provides the procedures for managing filesets, including creating, editing, renaming, copying and deleting.
Chapter 6: Scripts	Provides the procedures for creating, editing, renaming, copying and deleting scripts.
Chapter 7: Software Distribution Script Language	Lists the variables and rules for each function in the BrightWorks scripting language.
Chapter 8: Scopes	Provides the procedures for creating, editing, renaming, copying and deleting scopes; also discusses creating and managing queries to assist in scope definition.
Chapter 9: Packages	Provides the procedures for managing packages, including creating, editing, renaming, copying and deleting.

Chapter 10: Monitoring Software Distribution	Provides the procedures for viewing and managing the Software Distribution Log History.
Appendix A: Error Messages	Provides error messages for script editing, software distribution and the software distribution update program.

About BrightWorks

BrightWorks empowers network administrators to manage their networks more effectively. By offering software metering, asset management, software distribution, help desk and server monitoring capabilities, BrightWorks streamlines your network support efforts.

BrightWorks' modular design allows you to create the solution that best fits your network management needs by offering three integrated functions as well as access to other McAfee management solutions. In addition, BrightWorks' open framework facilitates integration with many management systems and products.

With years of networking experience, only McAfee can deliver an integrated solution as comprehensive and indispensable as BrightWorks.

BrightWorks' Features

The following lists the features available with each of BrightWorks' capabilities:

Software Metering

- Full enterprise metering including:
 - license sharing and load balancing over IPX and TCP/IP
 - centralized enterprise administration
 - reporting of enterprise metering
 - enterprise application usage viewing
- Server-based metering for DOS, Windows, OS/2 and Macintosh programs without using a workstation agent
- Simple installation to avoid administrative burden

- Detailed reports containing the information you need to make important management decisions about your network software
- Graphic displays detailing software usage to monitor both those currently using an application and those waiting to access a metered application
- Flexible enforcement options to allow you to monitor usage without denying access to applications
- Suite metering for accurate enforcement of concurrent license agreements for suite applications such as Microsoft Office
- Real-time trustee rights granting tied to application usage (masking) to control access to sensitive or critical network applications
- Ability to meter and control access to applications based on Novell group membership and the time of day
- Option to allow VIP users access to applications regardless of license availability
- Queueback for metered applications to reserve licenses for users waiting in the queue

Asset Management

- Recording and detecting software and hardware configuration changes (file servers, PCs & Macs) to eliminate the need for manual inventory
- Inventory collection for multiple sites to maximize resource usage
- Alerting of configuration changes via cc:Mail, MHS, e-mail and paging notification to enable timely network support and provide an added level of security
- Auto-learning of new software to reduce the time required to input new applications
- Detailed vendor and warranty data tracking to keep records up-to-date and to inform purchase decisions
- Import capabilities to preserve compatibility with other products and databases
- Full support for Compaq Intelligent Manageability Structure

Software Distribution

- Automated distribution of system files, data files and software applications to eliminate “sneaker-net” for these functions
- Flexible and powerful scripting languages which allow you to customize distribution of system files, data files and software applications
- Ability to edit system files (e.g., CONFIG.SYS, AUTOEXEC.BAT) to enable global replacement and workstation-specific changes without visiting each workstation

Server Monitoring

- Monitoring of over 110 NetWare internals for proper settings
- Unlimited number of file servers concurrently monitored
- Graphic displays of file server statistical data in real-time
- SmartTune feature that responds dynamically to fluctuating network activity and can react to heavy loads with appropriate tuning activity

Help Desk Automation

- Simplified help desk operation
- Rapid diagnoses and solutions to support problems
- Comprehensive reporting

Environment

The following criteria must be met in order to run BrightWorks.

Server Requirements

- Network Operating System: Novell NetWare 3.X and 4.X (with bindery emulation only)
- Network Disk Space: 18 MB required; 25 MB recommended
- Btrieve Database Access: server based: BTRIEVE.NLM

Administrator Console Requirements

- Operating System: DOS 5.0 or greater

- User Interface: Microsoft Windows 3.1X in enhanced mode
- Btrieve Database Access: server based: BREQUEST.EXE 6.1 or greater
- CPU: 386SX or higher
- RAM: 4 MB
- Monitor: VGA or better

DOS Workstation Requirements

- Operating System: DOS 3.3 or greater
- Btrieve Database Access: server based: BREQUEST.EXE 6.1 or greater; local based: BTRIEVE.EXE
- RAM: minimum of 640K

Macintosh Workstation Requirements (for Metering and Inventory only)

- Macintosh System 7 or greater)

OS/2 Workstation Requirements (for Metering only)

- OS/2 v. 2.1 or greater

NOTE: BrightWorks operates on Novell NetWare via IPX/SPX and includes multi-user BTRIEVE 6.10e.

NOTES

Chapter 2 Installation

Chapter 1 introduced BrightWorks. This chapter describes the installation procedures for BrightWorks.

NOTE: If you are installing the BBS release, unzip the files into a directory on your local or network drive.

Before Installation

To install BrightWorks, you must:

- Be logged in to the network as a SUPERVISOR or equivalent
- Run Windows 3.1X in enhanced mode
- Have the following line in the [386Enh] section of your SYSTEM.INI file:
- network=*vnetbios, vnetware.386, vipx.386

The following file versions are recommended for BrightWorks:

- IPX version 3.10
- NETX version 3.26 or greater
- Windows version 3.1X in enhanced mode
- VIPX version 1.13
- NETWARE.DRV version 2.02 or greater
- VNETWARE.386 version 1.06 or greater

NOTE: If you are using ODI drivers instead of IPX, you must have the following:

- LSL version 1.2 (2.01 is recommended)
 - IPXODI.COM version 1.2 (2.1 is recommended)
-

The latest versions of these files can be found on Compuserve in the Novell Libraries (GO NOVFILES). Current IPX, NETX, and IPXODI files are contained within the self-extracting files named VLMUP2.EXE and NET33X.EXE. Detailed information regarding these changes are located in DOSUP9.TXT.

Current versions of the Novell support drivers for Windows (VIPX.386, VNETWARE.386, NETWARE.DRV, etc.) are no longer contained in the self-extracting file WINUP9.EXE. WINDR2.EXE and NWDLL2.EXE have replaced the WINUP9.EXE file, detailed information regarding these changes are located in WINUP9.TXT.

NOTE: As these drivers are updated and added to the Compuserve file, the number within the Compuserve filename will increment. For example, if Novell releases a newer IPX and adds it to WINDR2.EXE, the name will change to WINDR3.EXE.

Determining Version Numbers

You can determine the versions of the above software by using the following methods:

- For IPX and the NETX shell versions, use the Novell NVER command.
- For Windows version and mode, run Windows and choose the Help | About Program Manager.
- For both Novell Windows support drivers and IPXODI.COM file versions, use the Novell VERSION command. For example, type:

```
VERSION VNETWARE.386 <ENTER>
or
VERSION IPXODI.COM <ENTER>
```

BrightWorks Installation

Installing BrightWorks is quick and simple, requiring minimal user input.

Use the following procedure to install BrightWorks on your network. You can exit the installation at any time by choosing Exit in the lower right corner of the installation screen.

During installation, BrightWorks modifies your existing WIN.INI file and backs up the old file as WIN.MCF. This change does not affect your Windows performance.

Refer to Appendix C in *Using LAN Inventory* or *Using SiteMeter* for information about installing and configuring BREQUEST.EXE and the Btrieve NLM. (The NLM and BREQUEST.EXE are required both to run the BrightWorks console and to perform the upgrade install.)

1. Verify that you have a drive letter mapped to the SYS: volume for the file server on which you are installing BrightWorks.
2. Start Windows.
3. Place the first distribution diskette in your floppy drive if you are installing from diskettes.

If you are installing from a Compact Disc (CD), place the CD in your CD drive.

If you are installing the BBS release, decompress the zipped files into a directory on your local or network drive.

4. Choose File | Run from your Windows Program Manager.

The Run dialog box is displayed.

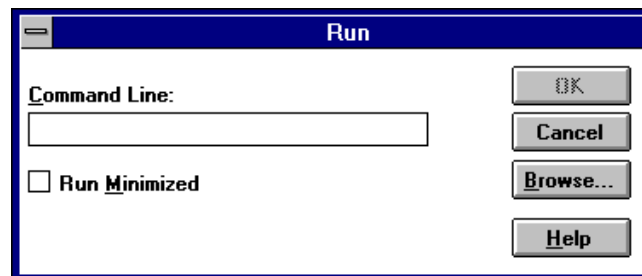


Figure 2-1: Run dialog box

5. At the prompt, enter the drive letter of the floppy, CD, or hard drive where you inserted the distribution diskette, CD, or where you unzipped the program files from the BBS and then type SETUP.

For example, type:

A:\SETUP <ENTER>

or

F:\BWKSDISK\SETUP <ENTER>

At this point, a message is displayed informing you that “Setup is initializing.”

NOTE: A log file is created and placed in your WINDOWS directory. The log file is an ASCII file listing the location of the BrightWorks installation. The log file also lists any errors that occurred during installation. If an error that prevents completion of the installation process occurs, the log file will display automatically.

The Welcome dialog box is displayed.

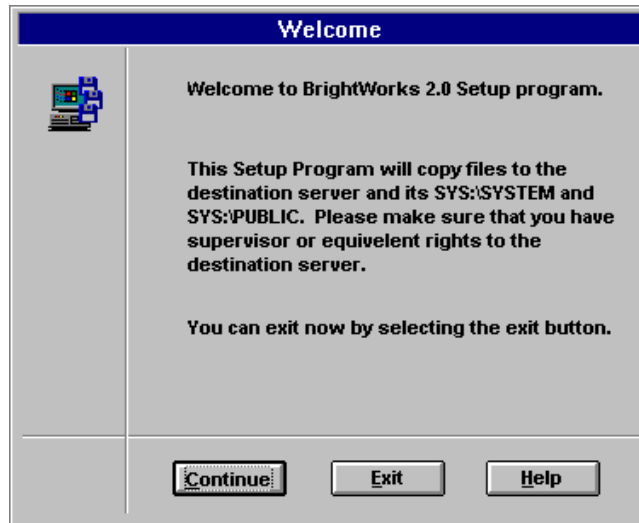
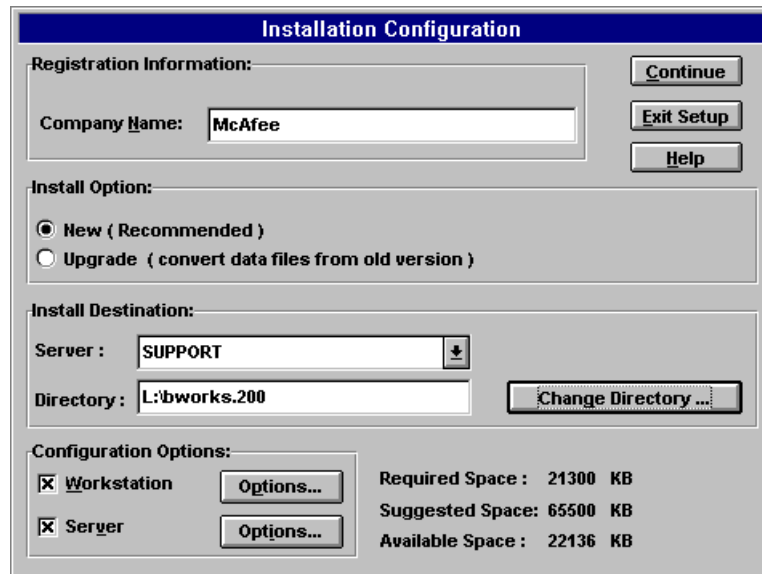


Figure 2-2: Welcome dialog box

6. Choose Continue.

The Installation Configuration dialog box is displayed.



The image shows a Windows-style dialog box titled "Installation Configuration". It is divided into several sections:

- Registration Information:** A text box labeled "Company Name:" contains the text "McAfee". To the right are three buttons: "Continue", "Exit Setup", and "Help".
- Install Option:** Two radio buttons are present. The first is "New (Recommended)" and is selected. The second is "Upgrade (convert data files from old version)".
- Install Destination:** A "Server:" drop-down menu shows "SUPPORT". Below it, a "Directory:" text box contains "L:\bworks.200". To the right of the directory box is a button labeled "Change Directory ...".
- Configuration Options:** Two checked checkboxes are shown: "Workstation" and "Server". Each has an "Options..." button next to it. To the right of these options, space requirements are listed: "Required Space : 21300 KB", "Suggested Space: 65500 KB", and "Available Space : 22136 KB".

Figure 2-3: Installation Configuration dialog box

NOTE: The Installation Configuration dialog box displays both the required space needed to run the BrightWorks install and the available space on the current server. If there is insufficient space, you will have to choose a new destination (i.e., volume or file server).

7. Type your company name in the Company Name text box.
8. Select one of the following install options:

Option	Description
New Install	Copies BrightWorks files to the network and automatically creates the Program Manager group McAfee (if not found) containing the BrightWorks program, the Crystal Reports program and all associated Readme file icons.
Upgrade	In addition to New Install features, it allows you to automatically upgrade from either previous versions of BrightWorks with minimal user input.

9. Select a server from the server drop-down list box.

The drop-down list box displays all the file servers to which you are currently attached and have a drive mapped. BrightWorks verifies that you have SUPERVISOR rights on the selected file server.

10. Confirm the Directory in the Directory text box.

The drive letter and full directory must coincide with the file server you selected earlier. BrightWorks creates the directory if it does not exist. The default drive letter is the first one found on the server you specified. BWORKS.200 is the default directory.

11. If you want to change the directory, choose Change Directory.

The Change Directory dialog box is displayed.

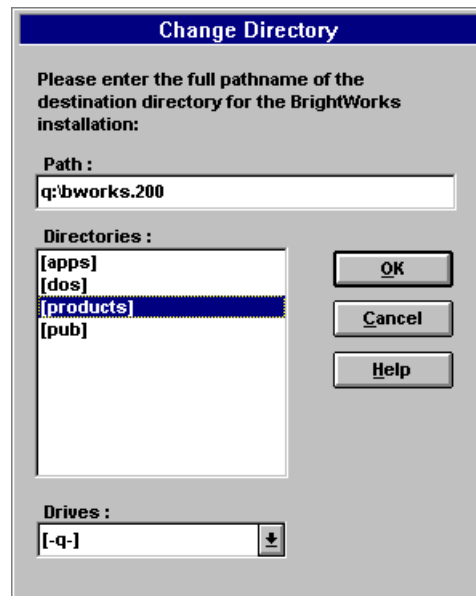


Figure 2-4: Change Directory dialog box

Select the desired directory and path and choose OK to return to the Installation Configuration dialog box.

12. If desired, deselect the Configuration Options.

By default, the install procedure configures both your workstation and server. Simply select the checkbox to disable these settings.

If you do select the Workstation Options button, the Workstations Configuration Options dialog box is displayed.

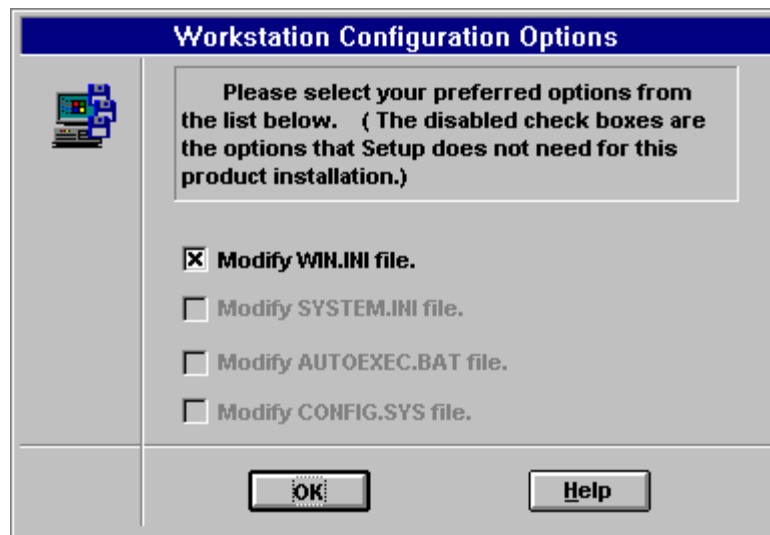


Figure 2-5: Workstation Configuration Options

If you do select the Server Options button, the Server Configuration Options dialog box is displayed.

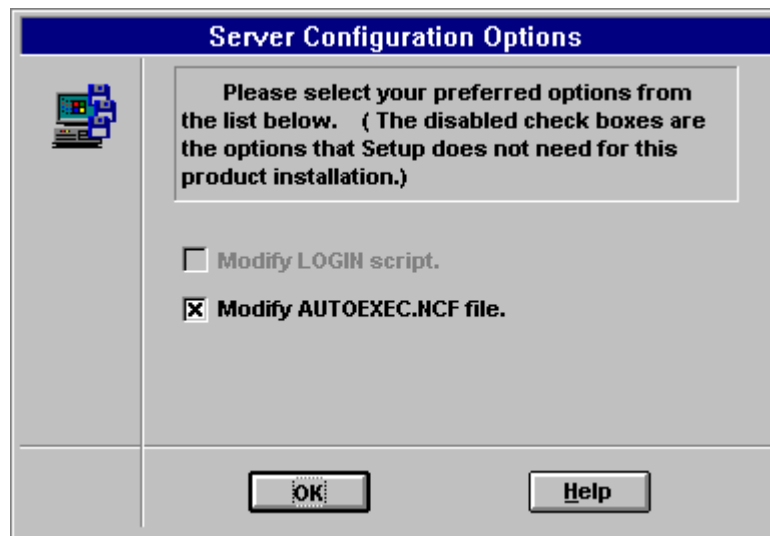


Figure 2-6: Server Configuration Options

Select the desired options in either dialog box and choose OK.

13. Choose Continue to proceed with the installation.

A dialog box is displayed with a percent completed bar.

14. If you selected the Upgrade option, refer to the next section for additional instructions.

If prompted, insert the remaining disks to complete the installation.

The Setup Information dialog box is displayed.

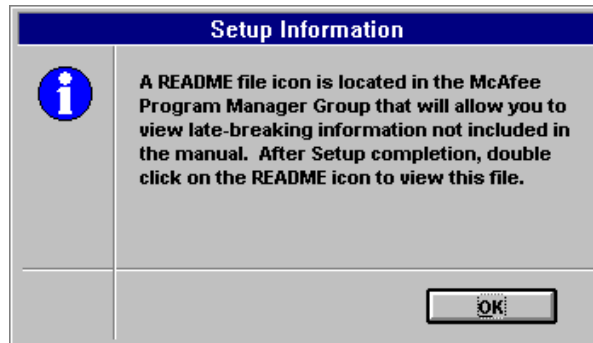


Figure 2-7: Setup Information dialog box

15. Choose OK to indicate that the installation is complete. View the Readme file for any updated product information.

The installation is complete. Refer to Chapter 3, "Getting Started" for a description of BrightWorks's console and tutorials introducing BrightWorks's metering capabilities.

Upgrade Install

The Upgrade option installs into a new BrightWorks directory and imports data from an old version of metering software. The term "migration" refers to the process of upgrading previous metering software with this version of BrightWorks.

Continue with the following steps to complete the Upgrade Install procedure:

15. After choosing Continue, the BrightWorks Upgrade Options dialog box is displayed prompting you to enter previous product installation location(s).

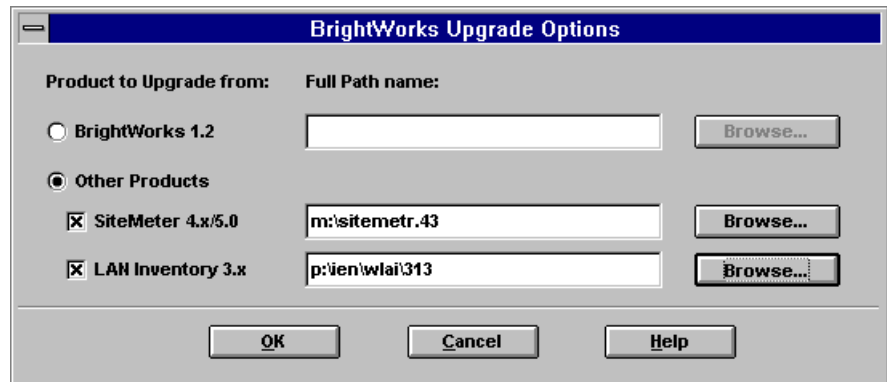


Figure 2-8: The BrightWorks Upgrade Options dialog box

Select either the BrightWorks or Other Products radio button and enter the previous installation(s) location in the provided text box. Choose Browse to display the Change Directory dialog box.

16. If prompted, insert the remaining disks to complete the installation.

The Confirm Migration dialog box is displayed and contains the following information:

Option	Description
Metering data directory	Displays the directory path where metering files are located.
File server to be migrated	Displays the file server that you entered in the Installation Configuration dialog box.
Disk space required	Displays the estimated amount of disk space needed to achieve a successful migration.
Disk space available	Displays the amount of disk space available on the selected server.

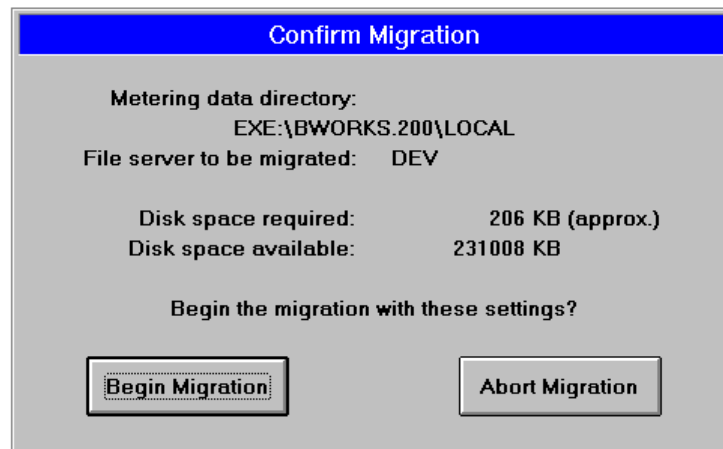


Figure 2-9: Confirm Migration dialog box

17. Choose Begin Migration to transfer the metered applications from previous copies to the selected file server.

The Migration Status dialog box is displayed.

NOTE: If you do not want to transfer the existing metered applications to the selected file server, choose Abort Migration. Your upgrade will install successfully, however, no metered applications will be transferred from previous copies of McAfee software metering products.

18. View the migration program.

The Success dialog box is displayed.

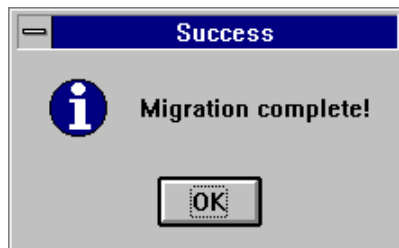


Figure 2-10: Success dialog box

19. Choose OK.

The Setup Information dialog box is displayed.

20. Choose OK.

The installation is complete. Refer to Chapter 3, “Getting Started” for a description of BrightWorks’s console and tutorials introducing BrightWorks’s metering capabilities.

Configuration Procedures

These instructions review the installation process and provide important information about installing the product NLMs.

Important: BREQUEST.EXE version 6.10 or later must be properly loaded to run the BrightWorks Administration program and USAGE.EXE. For instructions on setting up server-based Btrieve, refer to Appendix C, “Using Brequest” in either *Using LAN Inventory* or *Using SiteMeter*.

NOVDB.INI file must exist in the Windows directory. For server-based Btrieve, the Local=NO line should exist in that file.

If you are using BrightWorks and receive the following error message:

```
BrightWorks: The Novell Brequester has not been loaded.
BrightWorks databases will not be optimized. To optimize
performance of this program, load the Btrieve.NLM on
your server, the BREQUEST.EXE TSR on your workstation
(with argument /d:17000), and restart Windows and
BrightWorks.
```

and to the best of your knowledge Btrieve has been properly loaded, read the following instructions and take the appropriate steps.

This error occurs if the BTRIEVE.NLM or the BSPXCOM.NLM is not loaded on the file server. Please take the following steps:

1. At the file server console, type BSTOP.

This unloads the NLMs.

2. Type BSTART.

This loads both BTRIEVE.NLM and BSPXCOM.NLM on the file server.

Loading NLMs

The BrightWorks NLMs need to be loaded to begin collecting inventory and metering network software. To do this, at the file server, type:

BWORKS

This will load the required Btrieve NLMs, the LI.NLM, SITEMETR.NLM, SMRPROXY.NLM, SMRPT.NLM and SMRENT.NLM.

To unload the BrightWorks NLMs, type:

STOPBWRK

This will unload the BrightWorks NLMs.

Refer to both *Using LAN Inventory* and *Using SiteMeter* for information on the individual NLMs.

Chapter 3 *Getting Started*

Chapter 2 described the BrightWorks installation and upgrade instructions. This chapter introduces and discusses the BrightWorks application window.

NOTE: For a metering tutorial, refer to Chapter 3 in *Using SiteMeter*; for an inventory tutorial, refer to Chapter 3 in *Using LAN Inventory*.

The BrightWorks Console

Windows Terms

As a Windows application, BrightWorks should be used with a mouse. The table below briefly defines several Windows terms regarding the use of the mouse and product windows.

Term	Description
Button 1	The selection or primary mouse button (usually the left button, but can be switched using the Control Panel).
Cancel	Choose Cancel to exit the current dialog box without saving any of the changes you made in the dialog box or without executing a command you chose in the dialog box.
Choose	Double-click the mouse button (or use a key combination) on an item to initiate an action. For example, “Choose the BrightWorks icon” should be interpreted as a double-click on the BrightWorks icon.
Click	Press the mouse button once.
Double click	Press the mouse button twice in quick succession.
Icon	A graphic representation of an executable or function.
Point	Position the cursor on the screen to rest on the desired item.
Property Sheet	Windows tab metaphor that locates related information in a single dialog box and allows easy navigation from tab to tab.

Spin Control	Arrows that increase or decrease the value displayed in the accompanying text box.
Scroll	Use the scroll bars and buttons to move through a list of items.
Select	Mark an item by clicking on it or by highlighting it with either key combinations or the mouse. For example, “Select the Include Path option” should be interpreted as click or highlight the Include Path item.

NOTE: The remainder of this manual assumes that you are familiar with Windows. Refer to your Microsoft Windows manual for information on the fundamental operating conventions of the Windows environment.

Accessing BrightWorks

After successfully installing BrightWorks, a McAfee Program Manager group and a BrightWorks program icon are created on your Windows desktop.

Use the following procedure to launch BrightWorks.

1. Load Brequest.

The BrightWorks console requires the use of server-based Btrieve.

2. Run Windows, and double click on the BrightWorks program icon.

The application window consists of the following items which are discussed in this section:

- The BrightWorks Menu Bar
- The BrightWorks Tool Bar
- Access to BrightWorks' On-Line Help

Exiting BrightWorks

Use the following procedure to end a BrightWorks session.

1. Choose File | Exit.

A dialog box is displayed prompting you to confirm the exit action.

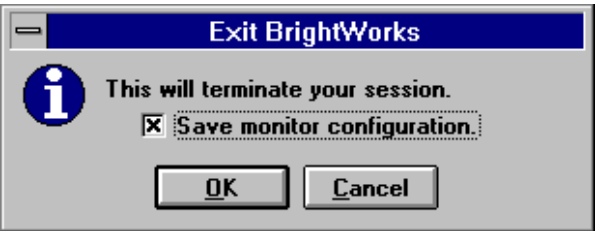


Figure 3-1: BrightWorks' Exit Prompt

2. Select 'Save monitor configuration.'

Checking this option will save the configuration of any metering windows that are currently open in your BrightWorks application window. All open metering windows will be automatically restored upon starting your next BrightWorks session.

3. Choose OK to close the BrightWorks application.

BrightWorks Menu Bar

The BrightWorks menu bar consists of the menu items listed in the table below. To choose a menu, point to the menu name and click the left mouse button. The menu displays.

Menu	SiteMeter	LAN Inventory	Software Distribution
File	New, Open, Close, Save, Delete, Print, Print Setup, Import, and Purge	New, Open, Close, Save, Delete, Print, Print Setup, Import, Purge	New, Open, Close, Save, Delete, Print, Print Setup, Rename, and Compile Script
Edit			Undo, Cut, Copy, Paste, Find, Next, Replace, Font, Paste Special
View	NLM Status, Hide Status Bar	NLM Status, Audit Log, Hide Status Bar	Refresh, Distribution Log,

			Hide Status Bar
Configure	Reporting, Prepare Report Data, Server System Settings, Set Refresh Timer, Edit Usage View	Audit Parameters, Alerting, Qualification Lists, Unidentified PC/Fileserver Software, Unidentified Macintosh Software	Distribution Preferences
Tools	Metered Applications, Replicate and Application Usage	Inventory	Filesets, QuickScripts, PowerScripts, Scopes and Packages
Reports	Choose Report, Edit Reports	Choose Report, Edit Reports	Choose Report, Edit Reports
Window	Cascade, Tile Vertically, Tile Horizontally, Arrange Icons, Close All	Cascade, Tile Vertically, Tile Horizontally, Arrange Icons, Close All	Cascade, Tile Vertically, Tile Horizontally, Arrange Icons, Close All
Help	Index, Using Help, About	Index, Using Help, About	Index, Using Help, About

NOTE: Holding down the left mouse button over a menu command causes the function of the command to display in the title bar at the top of the BrightWorks application window.

BrightWorks Tool Bar

When using BrightWorks with a mouse, BrightWorks' tool bar buttons provide an alternative for accessing the most frequently used BrightWorks functions. The tool bar is shown in Figure 3-2.



Figure 3-2: BrightWorks Tool Bar

Rather than choosing commands from the drop-down menus, you can choose the tool bar buttons to perform the same tasks. For example, to define applications to be

metered, you can either choose Tools | Metered Applications, or you can simply choose the Metering tool bar button. Both actions result in displaying the Define Metered Applications dialog box.

The function of each tool bar button is described below:

Button	Description
Meter	Displays the Define Metered Applications dialog box used for adding, modifying and deleting applications to be metered.
Usage	Displays the View Application Usage dialog box used for determining which applications are being used and by whom.
Inventory	Displays the Inventory dialog box used for viewing and managing the inventory of each audited workstation.
Distribute	Displays the Available Packages window used for viewing, creating and managing packages for distribution.
QuickScript	Brings up the QuickScript editor to write Software Distribution scripts.
PowerScripts	Brings up the DCL PowerScript editor to write flexible, powerful Software Distribution scripts.
NetRemote	Provides access to optional McAfee NetRemote software for automated user support.
SupportCtr	Provides access to optional McAfee LAN Support Center software for help desk automation.
NetShield	Provides access to optional McAfee NetShield software, which is a NetWare loadable module (NLM) that provides uninterrupted server-based virus protection.
NetTune	Opens the NetTune server monitoring and tuning application.
Alerts	Displays the Alerting Options dialog box used for defining and scheduling auditing alerts.
Reports	Displays the Choose Report dialog box, enabling you to generate inventory, metering and distribution reports.

NOTE: Holding down the left mouse button over a tool bar button displays the function of the button in the BrightWorks title bar at the top of the application window.

Using the Keyboard

To use BrightWorks without a mouse, perform the standard Windows keyboard actions to navigate through the program.

Each menu item on the BrightWorks menu bar has a keyboard mnemonic. Press the <ALT> key in combination with the keyboard mnemonic key to choose a menu and cause the menu to drop down. For example, press the <ALT><F> keys to choose the File menu and display its commands.

Each command also has a keyboard mnemonic. Once the menu is displayed (i.e., “dropped down”), press the keyboard mnemonic of the command you want to choose. For example, from the File menu, press <P> to choose the Printer Setup command. You can also use the <up/down arrow> keys to move the highlight to a desired command and press <ENTER> to select the command.

For detailed information on using a Windows application with the keyboard, refer to your Microsoft Windows documentation.

NOTE: Some BrightWorks features require the use of a mouse and cannot be accessed with the keyboard.

BrightWorks’ Help Facility

BrightWorks’ help facility provides on-line assistance for using the BrightWorks software. To get information quickly about a BrightWorks feature or procedure, choose Help | Help Index.

Choosing the Help Index command displays an index list of topics. Choose the topic for which you require assistance.

BrightWorks’ Help system is written in a standard Windows hypertext format. This allows you to jump from one topic to another by simply choosing topic names from a list. Several buttons display across the top of the Help dialog box allowing you to search for topics and also to view a list of the topics you have visited.

For detailed information on using a Windows help facility, refer to your Microsoft Windows documentation.

BWORKS.INI File

Upon installing the BrightWorks software, the BWORKS.INI file is created and placed in the Windows directory of the local workstation. The file can consist of the following sections:

- **[ShowBrequestWarning]** - This section indicates the status of the Brequest warning which displays upon launching BrightWorks when the presence of the Btrieve NLM is not detected. This section contains an “Init=” line which indicates whether or not the warning is disabled (i.e., “Init=No” when warning is disabled; “Init=Yes” when warning is enabled).

The warning message can be disabled from within BrightWorks by checking the “Disable message when Brequest isn’t running” field when launching the application. To re-enable the warning message after it has been disabled, edit the BWORKS.INI file, and enter “Init=Yes” in this section.

- **[ShowBanner]** - This section can be manually added to the INI file in order to disable the About BrightWorks dialog box which displays upon launching BrightWorks. To disable the About dialog box at start-up, enter the following:

```
[ShowBanner]
```

```
Init=No
```

To re-enable the warning message, either delete this section or enter “Init=Yes.”

- **[DisableExitPrompt]** - This section can be manually added to the INI file in order to disable the prompt which displays upon exiting BrightWorks. This prompt allows you to save the configuration of any open metering windows. (Refer to Figure 3-1 in the section above entitled “Exiting BrightWorks.”) To disable the exit prompt, add the following section to the BWORKS.INI file:

```
[DisableExitPrompt]
```

```
AutoSave=Yes
```

Note that an “AutoSave=Yes” setting will inhibit the exit prompt and save the configuration of the open *metering windows*. An “AutoSave=No” setting will also inhibit the exit prompt but will not save the open metering window settings.

- **[Alternate EXE]** - This section can be manually added to the INI file in order to change the applications that are launched when the BrightWorks NetRemote or SupportCtr tool bar buttons are pressed. For example, to run a third party executable help desk program when the SupportCtr tool bar button is chosen, add the following section to the BWORKS.INI file:

```
[Alternate EXE]
```

```
supportctr=3rdparty.exe
```

To launch another program when the BrightWorks NetRemote tool bar button is chosen, add the line “remote=” to this section and indicate the alternative application’s executable file name. In all cases, to successfully launch another program from within BWORKS, the program must be in the Windows directory, Windows system directory, DOS path or search path.

- **[downgrade]** - This option allows you to instruct the console to display either metering or inventory functions when it displays. For only metering functions, add the following section to the BWORKS.INI file:

```
[downgrade]
module=metering
```

To display only inventory functions, add the following section to the BWORKS.INI file:

```
[downgrade]
module=inventory
```

Printer Setup and Administration

Before printing BrightWorks reports, review the global print parameters to ensure they reflect the printer settings that you require.

Printer settings include:

- Printer destination
- Page orientation (portrait/landscape)
- Paper size and source
- Graphics resolution

The procedures for customizing the contents of individual BrightWorks reports are discussed in Chapter 9 of *Using LAN Inventory* (for inventory and distribution reports) and Chapter 7 of *Using SiteMeter* (for metering reports). This section briefly presents the procedures for viewing and changing Windows global print settings (e.g., target printer, paper size).

NOTE: Please refer to your Microsoft Windows manual for detailed procedures on modifying the Windows print settings.

Changing Print Settings

Use the following procedure to review and change your print settings.

1. Choose File | Print Setup.

The Print Setup dialog box is displayed.

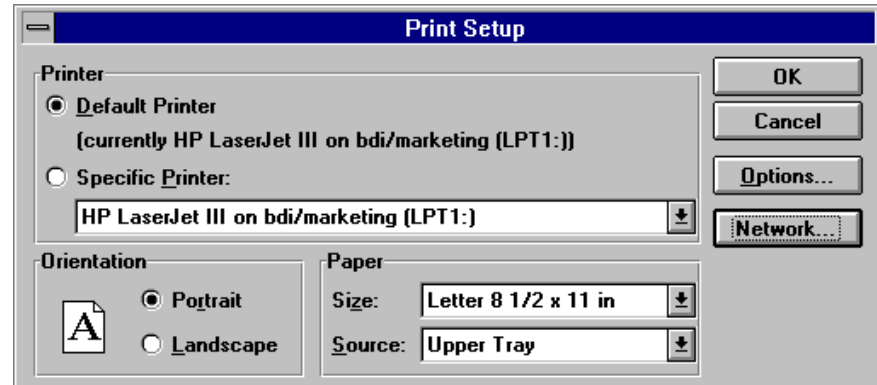


Figure 3-3: Setting Print Parameters

If you have the latest version of COMMDLG.DLL, this dialog box will have a Network button.

2. Select the printer you want to use for printing LAN Inventory reports.

The printer selected from your Windows printer control is selected as the default. To use another printer, select a Specific Printer from the drop-down list associated with this field.

NOTE: Choosing a specific printer does not permanently change your printer setting.

3. Select the desired orientation and paper parameters.

Choose either the Portrait (long) or Landscape (wide) Orientation radio button. Use the drop-down lists to define the Paper Size and Paper Source settings.

4. To make additional changes to the selected printer configuration, choose Options.

Additional settings include dithering and intensity control.

5. Choose OK in the Print Setup dialog box to save the print settings.

Distribution Tutorial

This tutorial outlines the major steps in using BrightWorks to distribute software and scripts throughout your local area network. Refer to Chapter 3 in *Using LAN Inventory* for an inventory tutorial and Chapter 3 in *Using SiteMeter* for a metering tutorial.

NOTE: BrightWorks' distribution component must be installed on your network before beginning the tutorial. If you have not already done so, please refer to Chapter 2, "Installation."

The steps in this tutorial include the following:

1. Creating and Compiling a Script
2. Creating a Scope
3. Scheduling a Package for Distribution
4. Running the SDUPDATE.EXE Program at the Receiving Workstation.

NOTE: The options and features mentioned in this tutorial are discussed in detail throughout *Using BrightWorks*.

Creating and Compiling a Script

A script is a series of commands to be executed on a remote workstation. Scripts can be as simple as displaying a message on a remote user's monitor, or they can be as complex as restructuring a hard drive and/or installing and configuring software.

Use the following procedure to create a script.

1. Choose Tools | QuickScripts.

The Scripts dialog box is displayed. For each script you define, the last compilation date, the status and the file name will display.

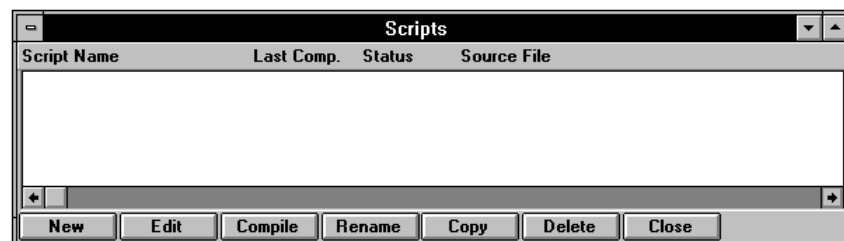


Figure 3-4 The List of Scripts

2. Choose New.

The Open New Script dialog box is displayed prompting you to enter the name, file name and destination directory for the new script.

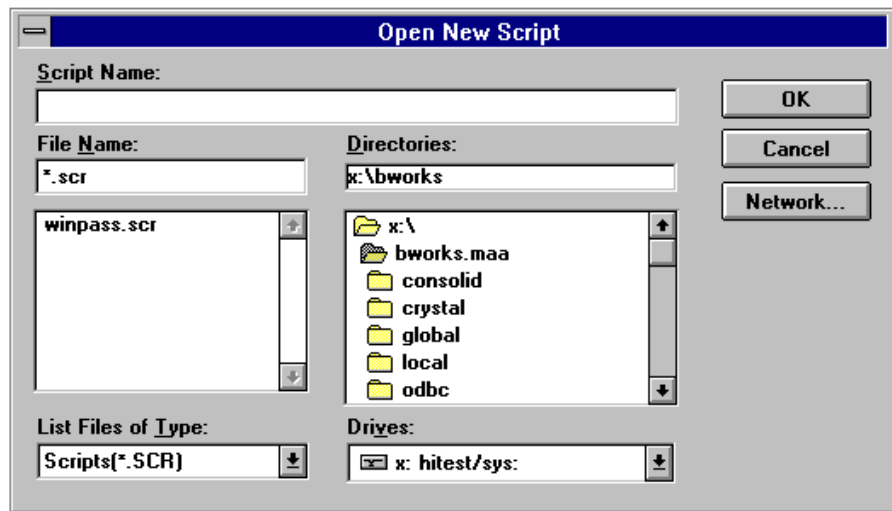


Figure 3-5: Creating a New Script

NOTE: The Network button will only appear in dialog boxes if you have the latest version of COMMDLG.DLL.

3. In the Script Name text box, enter HELLO WORLD.
4. In the File Name text box, enter Hello.SCR.
5. Choose OK.

The message “This file does not exist. Create the file?” displays.

6. Choose Yes to create the script file.

The Script editor window automatically displays so you can enter the script commands. The script name being edited displays in the title bar of the script editor dialog box.



Figure 3-6: A New Script Editor Window

7. For the purpose of this tutorial, enter the following script command:

```
WRITELN "HELLO WORLD! "
```

The WRITELN script function is used to write a value (in this case, “Hello World”) to the screen at the receiving workstation.

NOTE: The commands and rules for using the scripting language are documented in Chapter 7, “Software Distribution Script Language.”

8. Choose Save.

The script is saved and the editor window remains open. The saved script contents are stored in ASCII text format.

NOTE: Scripts must be compiled before they can be used in a package. Continue with the following steps to compile the script.

9. Choose Compile.

While a compile is in progress, the Compile Status dialog box is displayed.

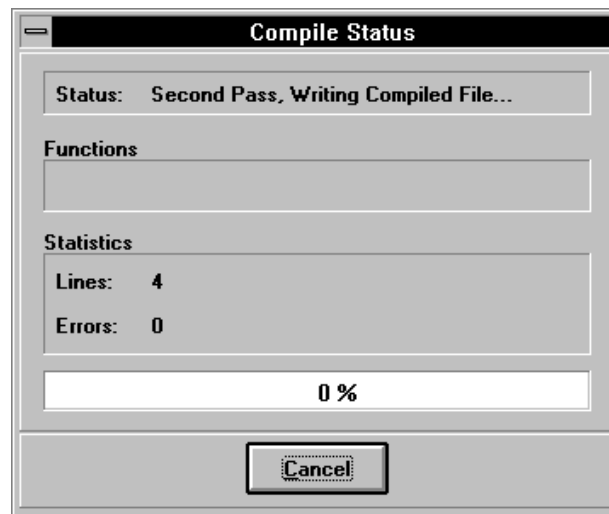


Figure 3-7: Script Compilation

When the compile is complete, the Status field in the Compile Status dialog box indicates success or failure. If the compile fails, the Function field indicates the first function found with invalid parameters. The Statistics area indicates the total number of lines in the script (Lines field) and the number of errors found (Errors field).

If you entered the script command exactly as shown in Step 5 above, your script should compile successfully.

10. Choose OK to continue.

The Compile Status dialog box closes.

11. Choose Close in the Script Editor dialog box to return to the Scripts dialog box.

The script's status displays as **COMPILED**. A script can be used in a package only after it has been compiled successfully.

12. Choose Close in the Scripts window.

Creating a Scope

A scope is a group of workstations defined to receive a distributed package. Any number of workstations can be included in a scope definition.

Use the following procedure to create a new scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed.

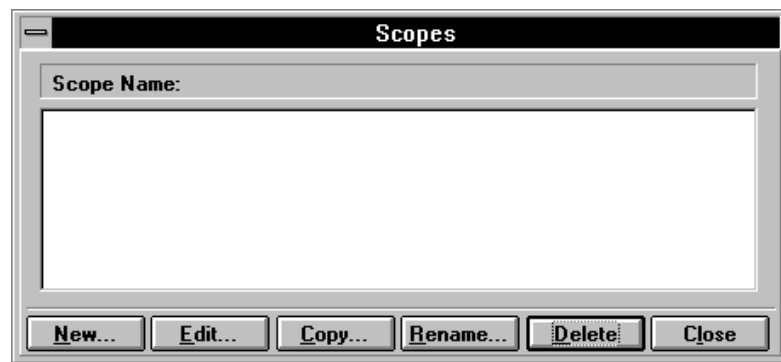


Figure 3-8: The List of Scopes

2. Choose New.

The New Scope dialog box is displayed prompting you to enter a name for the new scope.

3. In the Scope Name text box, enter TUTORIAL_SCOPE.
4. Choose OK.

The Edit Scope dialog box is displayed prompting you to define the new scope, similar to Figure 3-9.

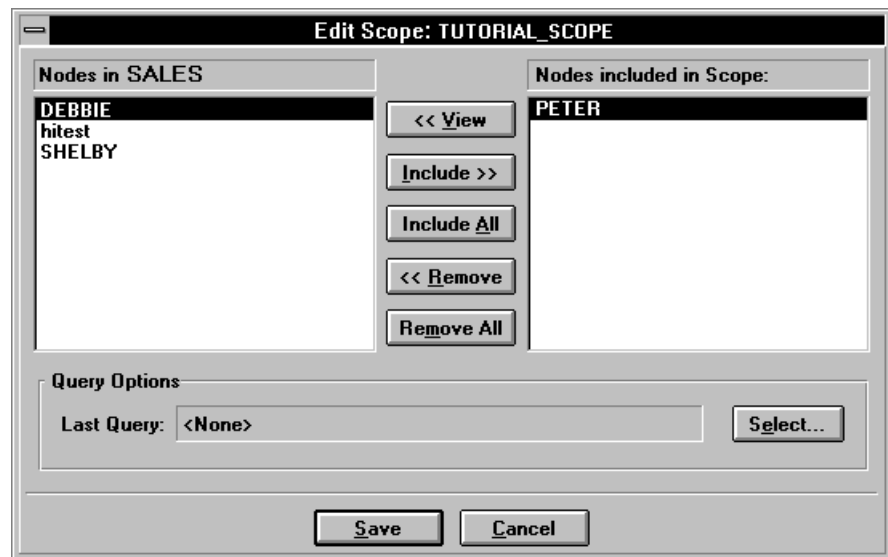


Figure 3-9: Editing a Scope

5. Choose Include All.

NOTE: The Query Options section of the Edit Scope dialog box is used to perform a query to filter the node names in the Nodes in SITE list. Refer to Chapter 8, "Scopes" for further information.

6. Choose Save.

The scope's membership is defined, and you are returned to the Scopes dialog box.

7. Choose Close.

Scheduling a Package

Software is distributed across your local area network by creating and activating a package. When a package is created, it is assigned a scope and a "Start Date." Upon reaching the start date and running the SDUPDATE.EXE program at a workstation in the scope, an active package is automatically sent to the workstation.

Use the following procedure to schedule a package consisting of the scope and script created in this tutorial.

1. Choose Tools | Packages.

The Packages dialog box is displayed. This dialog box lists the names and status information of all defined packages.

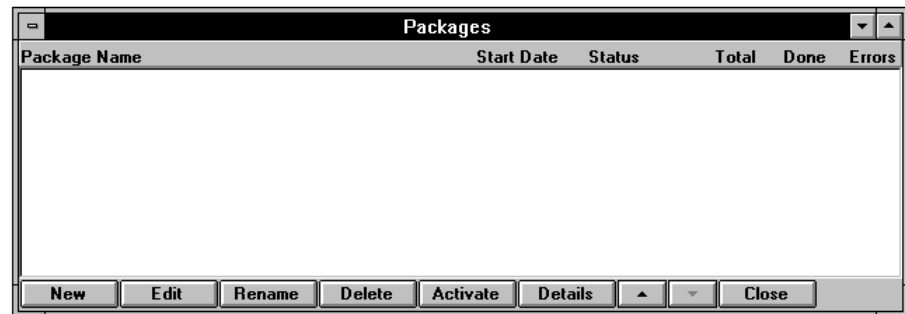


Figure 3-10: The List of Packages

2. Choose New.

The New Package dialog box is displayed prompting you to enter a name for the new package.

3. In the text field, enter TUTORIAL.
4. Choose OK.

A New Package dialog box is displayed. The name of the new package is indicated in the title bar of the dialog box.

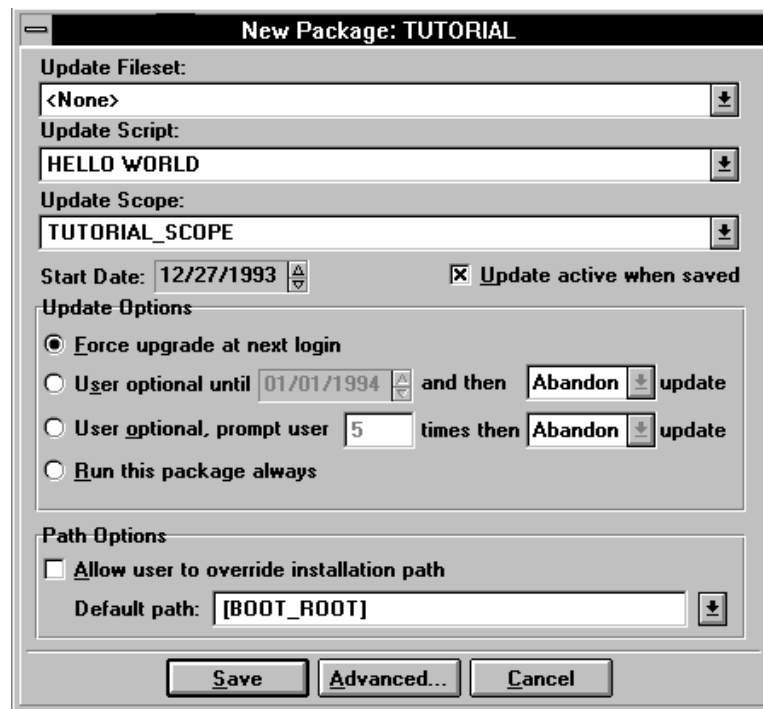


Figure 3-11: Creating a New Package

5. Select Hello World from the Update Script drop-down list box.

NOTE: A package must include a fileset and/or a script.

6. Select TUTORIAL_SCOPE from the Update Scope drop-down list box.
7. Select the Update active when saved option.

This option automatically places the package in an active state upon saving the package. (An active package will get distributed automatically on its assigned start date.)

8. In the Upgrade Options group box, select "Force upgrade at next login."

This option forces the package's receipt on the user at the next login. If an error occurs, the distribution is halted so you can address the problem and reschedule the package.

9. In the Path Options group box, set the Default Path to [BOOT_ROOT].

Even though this package does not consist of a fileset, a default path must be assigned to the package. The default path is the target path in which the distributed software (e.g., fileset) is to be installed or copied.

10. Choose Save.

The New Package dialog box closes, and the package is saved and assigned an active status.

Running the Update Program

The distribution update program (SDUPDATE.EXE) is a DOS-based program which must be run from each workstation in order to receive the distributed packages it has been sent. Upon BrightWorks installation, the update program is copied into the BWORKS.200 directory.

```
SDUPDATE [drive:[\path]]
```

in which *drive* and *path* are optional parameters. The brackets are not typed.

Consider the following examples:

Example	Result
SDUPDATE	SDUPDATE will look in the current directory.
SDUPDATE F:	SDUPDATE will look in the current directory on drive F:.
SDUPDATE F:\path	SDUPDATE will look in the directory F:\path.

Use the following procedure to run the update program and distribute the active package.

1. At the workstation which is to receive the distributed package, load the Btrieve Record Manager.
2. Use the DOS CD command to change into the BWORKS.200 directory.
3. At the DOS command line, type: SDUPDATE.

Upon executing SDUPDATE, messages similar to the following will display at your workstation:

```
Looking for packages, please wait...
Installing package "TUTORIAL"...
Created target path.
Target path: C:\
Disk space: ### bytes free
Running Script "HELLO_WORLD"...
HELLO WORLD!
The script completed successfully.
No more packages scheduled.
```

This completes the distribution tutorial. All of the features introduced here are described in full detail in rest of this manual.

Chapter 4 *Software Distribution*

Chapter 3 described the BrightWorks console and provided a brief tutorial. This chapter introduces the distribution feature.

About the Software Distribution Capability

BrightWorks' software distribution capabilities provide a method for distributing software packages and modifying workstation configuration files from a central location. The software distribution features facilitate consistency among the workstations across your local area network and improve the productivity of the LAN Administrator.

Distributing software and modifying workstation configuration files from a central location on your network allows the LAN Administrator to easily do the following:

- update system executables and/or drivers (e.g., operating systems, network drivers)
- update system files (e.g., AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, network login script)
- install and update software on user workstations across the local area network

NOTE: BrightWorks' Software Distribution capabilities can be used to distribute software and/or scripts to any workstation in the BrightWorks local site (i.e., the site which identifies the BWORKS.200 program directory). For further information, refer to "Maintaining Network Sites" in Chapter 4 of your *Using LAN Inventory* manual.

Software Distribution Concepts

An understanding of the following concepts will help you in gaining full advantage of BrightWorks' software distribution capabilities:

Item	Description
Fileset	A file that contains one or more compressed files. Each compressed file may also indicate a target directory structure in which the file should be decompressed. For example, assume a fileset named NEW_INI_FILES. The fileset might consist of two files: WIN.INI and SYSTEM.INI which have been defined to be decompressed into a target directory named PUB\WIN.310.
Script	A sequence of one or more commands which define an operation to be performed on a workstation receiving a distribution. For example, a script might include the commands to add a new group to the Windows Program Manager, to copy file(s) from one location to another, or to change parameters within certain files.
Scope	A group of one or more workstations that have been identified to receive a distribution. For example, to distribute a script to all 386 workstations, you must create a scope which includes the 386 workstations.
Package	The distributed object which contains scheduling information, as well as a fileset and/or script and a scope.

Figure 4-1 illustrates the creation of a package.

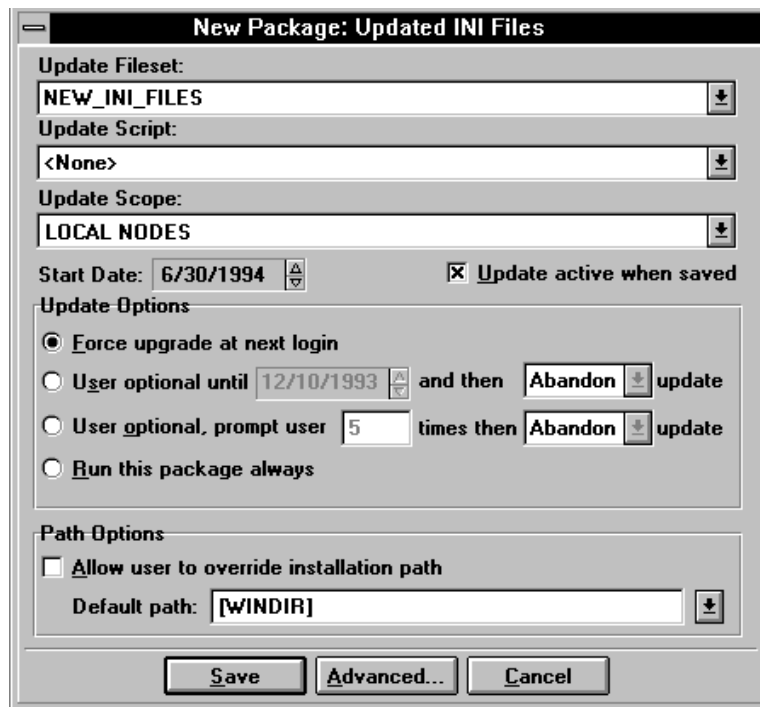


Figure 4-1: Creating a New Package

In Figure 4-1, the package being created is named “Updated INI Files,” as indicated in the dialog box title bar. The package is scheduled to be distributed on 6/30/1994. The package consists of the fileset named “NEW_INI_FILES,” which will be distributed to all nodes included in the scope named “LOCAL NODES.”

The following key features aid you in distributing software across your network:

- Creating *filesets* which include files to be installed on remote workstations.
- Creating *scripts* to be executed by remote workstations.
- Defining *scopes* of workstations to receive distributed packages.
- Creating and scheduling packages which consist of a scope and one fileset and/or one script.
- Monitoring package progress through the Software Distribution Log History dialog box.

BrightWorks' Software Distribution Modules

The BrightWorks software distribution capabilities interact with two major functional modules. As an introduction to software and script distribution, this section briefly describes the following modules:

- BrightWorks console and administrative functions
- Remote workstation update program (SDUPDATE.EXE)

NOTE: BrightWorks' software distribution capabilities are provided with the inventory capabilities discussed in *Using LAN Inventory*.

BrightWorks Console/ Administrative Program

BWORKS.EXE is the BrightWorks console and administrative program. This program provides access to all BrightWorks capabilities. This main module is a Windows-based program and is intended to be used by the network manager to perform all software distribution functions.

The software distribution functions available from the BrightWorks console include:

- Scope definition and management
- Script creation and management
- Package creation, scheduling and management
- Pre-defined and custom distribution report generation

Update Program

The update program (SDUPDATE.EXE) must be executed from each remote workstation to enable the workstations to receive the distributed packages they have been sent. Upon BrightWorks' installation, the update program is copied into the BWORKS.200 program directory.

The update program is DOS-based and must be executed from the machine which is to be updated. To ensure that SDUPDATE.EXE is executed on a regular basis, the command can be placed in the system login script. Refer to “Distribution Configuration Options” in this chapter for further information.

NOTE: WSDUPD.EXE is the BrightWorks update program which handles the script functions related to installing Windows software. This program must not be directly run by the user—it is automatically loaded when the ADDGROUP, ADDITEM or SCHEDULEWIN Windows System File script functions are used. For further information, refer to “Windows System File Functions” in Chapter 7.

Distribution Configuration Options

The update program SDUPDATE.EXE must be run from each workstation in order for it to receive the distributed packages it has been sent. Upon BrightWorks' installation, the update program is copied into the BWORKS.200 program directory.

The SDUPDATE program's syntax is as follows:

```
SDUPDATE [drive:[\path]]
```

in which *drive* and *path* are optional parameters. The brackets are not typed.

Consider the following examples:

Example	Result
SDUPDATE	SDUPDATE will look in the current directory.
SDUPDATE F:	SDUPDATE will look in the current directory on drive F:.
SDUPDATE F:\path	SDUPDATE will look in the directory F:\path.

The Btrieve Record Manager must be loaded before running SDUPDATE.EXE.

When running the Brequestor, BSPXCOM must also be loaded on the file server. For details on loading these programs, refer to your Novell documentation.

When running SDUPDATE.EXE in a DOS box from within Windows, you must load another session of BREQUEST by entering the following command:

```
BREQUEST /D:17000 /L
```

After running SDUPDATE, end the additional session by issuing the ENDBTRV command.

Running the Update Program

Use the following procedure to manually run the update program at a workstation.

1. At the workstation which is to receive the distributed package, load the Btrieve Record Manager.
2. Use the DOS CD command to change into the BWORKS.200 program directory.

Otherwise map a physical drive to the BWORKS.200 directory.

3. Execute the SDUPDATE.EXE program.

For example, enter the following at the DOS command line:

```
SDUPDATE
```

Upon executing SDUPDATE, several messages will display at the workstation. If the user has not been given the option to refuse the update or change the installation path, then the update program will continue automatically (e.g., the package's script or fileset will be installed at the workstation).

4. If you are given the option of refusing the package, then the prompt illustrated in Figure 4-2 displays.

To install the package at this time, type <Y>. To install the package another time (e.g., the next time the update program is run), type <N>.

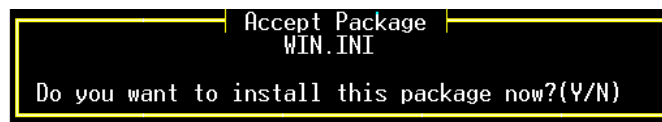


Figure 4-2: The Update Program's Install Option Prompt

NOTE: If the date or maximum number of times has expired and the package is configured to 'force upgrade,' then the package will be installed regardless of the user's response to this prompt.

5. If you are given the option of overriding the installation path, then the prompt illustrated in Figure 4-3 displays.

To override the default installation path, type <Y>. To accept the default installation path, type <N>.

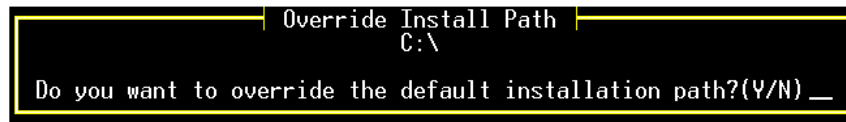


Figure 4-3: The Update Program's Installation Override Prompt

If you type <Y> to override the default installation path, you are prompted to specify a new installation path. Type the new path and press the <ENTER> key. The update program continues, and the package is installed.

Automating the Update Program

To ensure that SDUPDATE is executed on a regular basis, the command can be placed in the system login script.

The following example illustrates SDUPDATE being executed from within the system login script.

```
....
MAP G:=SERVER/SYS:BWORKS.200
DRIVE G:
#BREQUEST /D:17000
#SDUPDATE
#ENDBTRV
....
```

where *G:=SERVER/SYS:BWORKS.200* is the drive ID and complete path where the BrightWorks files and update program are stored.

For further information regarding the Btrieve NLM, refer to Appendix C in *Using LAN Inventory* or *Using SiteMeter*.

NOTES

Chapter 5 *Filesets*

Chapter 4 introduced BrightWorks' software distribution capability. This chapter discusses the creation and management of filesets—the set of files to be installed on a remote workstation.

Introduction

A fileset is a set of files stored in compressed format to be installed on a remote workstation. Distributing filesets from a central location simplifies network administration—instead of physically moving from workstation to workstation to install or upgrade application files, you only need to centrally distribute one fileset consisting of the application files. Upon receipt at a remote workstation, the fileset contents are decompressed and copied onto the workstation's hard drive.

Fileset Features

In addition to containing a number of files to be distributed, filesets can be defined to create a target directory structure. For example, if you create a fileset which includes all files for Windows 3.1, you must also define the contents of the SYSTEM subdirectory. BrightWorks can do this for you automatically by including the full path name of every file included in the fileset.

Filesets and scripts are a powerful combination. Consider the following examples:

- Packaging the latest WIN.INI file with a script which determines whether the existing WIN.INI file is outdated. The script will also copy the new WIN.INI if an old file is detected.
- Packaging the Novell IPXODI files and sending them to the scope of nodes using IPX. After the fileset is decompressed in the target directory, the script will update the AUTOEXEC.BAT file to reflect the use of IPXODI.

Filesets can be stored, used and reused as a resource within BrightWorks. An administrator can create a new fileset, as well as edit, copy, rename and delete a fileset. The steps for each procedure are provided in this chapter.

The Fileset Directory

The fileset directory defines the path in which filesets are stored. Upon saving a fileset, a copy of the files that are included in the fileset are compressed. They are stored in a file which is placed in the fileset directory defined at the time the fileset is saved. For example, if your fileset directory is defined as

F:\BWORKS.200\FILESETS, then the filesets that you create will be stored in the F:\BWORKS.200\FILESETS directory.

Defining a Fileset Directory

Use the following procedure to define the directory in which filesets should be stored.

1. Choose Configure | Distribution Preferences.

The Preferences dialog box is displayed.

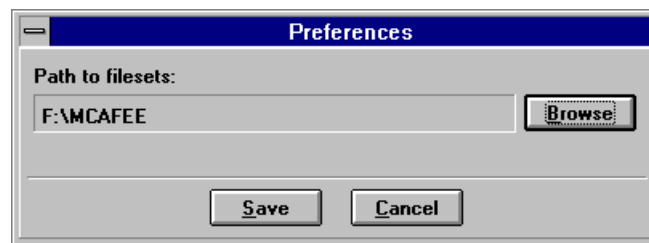


Figure 5-1: Defining the Path to Filesets

2. Choose Browse to define the pathname into which the compressed filesets are to be stored.

The Path Browse dialog box is displayed enabling you to select from the lists of Drives and Directories. Click on the Drives and Directories fields to select the desired pathname.

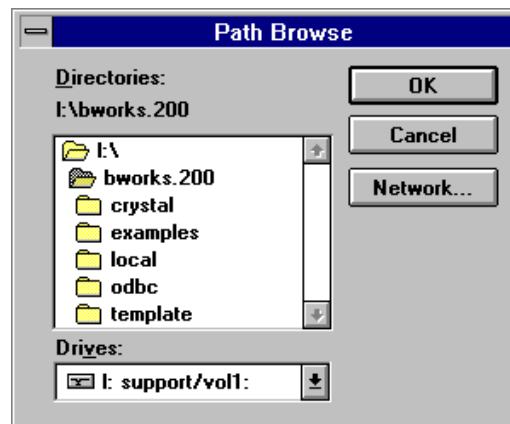


Figure 5-2: Browsing the Directory Tree

3. Choose OK.

The Path Browse dialog box closes, and the selected pathname displays in the Path to Filesets field of the Preferences dialog box.

4. Choose Save to define the fileset directory.

All saved filesets will be stored in the defined directory.

NOTE: The fileset directory instructs the update program as to where the filesets are located. As a result, changing the fileset directory after you have created filesets and included them in packages can invalidate those packages. If you change the default directory, you must also copy all fileset files (*.SET) into the new fileset directory. Ensure that each user has the same drive letter mapped to the same server/volume specified in the Preferences dialog box. Filesets cannot be stored on a non-network drive.

Creating Filesets

Use the following procedure to create a new fileset.

1. Choose Tools | Filesets.

The Filesets dialog box is displayed listing the names of all defined filesets.

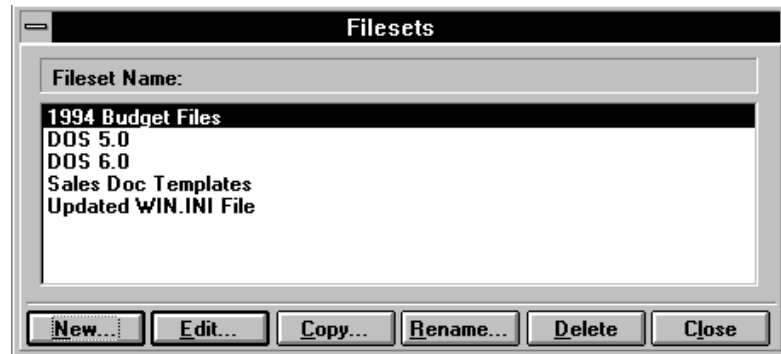


Figure 5-3: A List of Filesets

2. Choose New.

The New Fileset dialog box is displayed prompting you to enter a name for the new fileset.

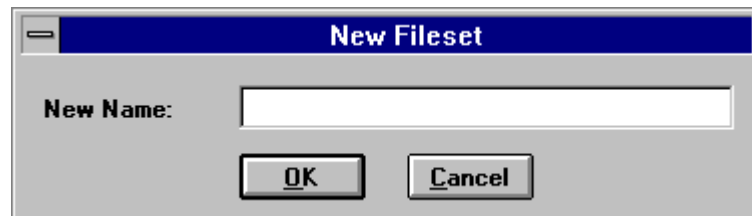


Figure 5-4: Adding a New Fileset

3. Enter the new fileset name.

A fileset name can be up to 80 characters, and all typed characters are valid.

4. Choose OK.

The Edit Fileset dialog box is displayed prompting you to define the contents of the new fileset.

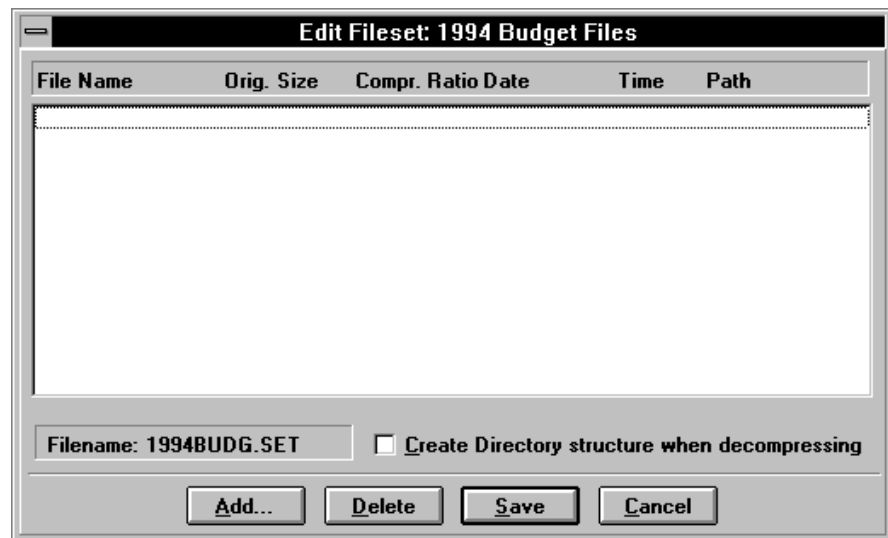


Figure 5-5: Creating a New Fileset

The fileset name being created or edited displays in the title bar of the Edit Fileset dialog box. In Figure 5-5, the name of the fileset being created is “1994 Budget Files.”

For each file included in the fileset, the following information displays: File Name, Original Size, Compression Ratio, Date, Time, and Path. The file list area is empty for new filesets.

The Filename field in this dialog box is displayed the name of the file which will hold the compressed fileset. In Figure 5-5, upon saving the fileset, a compressed copy of all of the listed files will be stored in the file named “1994BUDG.SET.” This file is automatically created by BrightWorks when the fileset is created. It is stored in the fileset directory which is currently defined.

5. Choose Add.

The Add File dialog box is displayed.

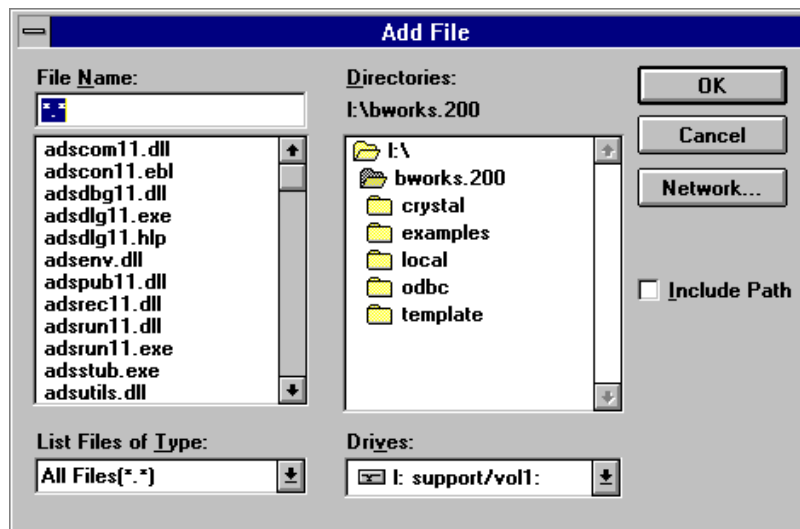


Figure 5-6: Adding Files to a Fileset

This dialog box is a standard Windows dialog box used for opening, selecting and browsing files.

6. Make selections from the Directories and Drives lists to find the file(s) to include in the fileset.
7. Select a file(s) from the File Name list.

Multiple files can be selected using the Windows extended select procedures (i.e., hold down the CTRL or SHIFT key while selecting files).

8. To include the selected file(s)' path in the Edit Fileset dialog box, select Include Path.

Placing a checkmark in this field will list the full pathnames of each selected file in the Edit Fileset dialog box. (Step #9 below provides the option to instruct the fileset to create the directory structure at the receiving workstation.)

9. Choose OK.

The selected files are listed in the Edit Fileset dialog box. Only the File Name and Path information display at this time; the other fields are not available until the fileset is saved.

10. Enable or disable the Create Directory Structure option.

Enabling this option causes the full pathnames of each file listed in the Edit Fileset dialog box to be created at the receiving workstation. For example, assume that this option is enabled and a file is listed in the Edit Fileset dialog

box as \USER\MARY\INVOICE.DOC. In this case, the directories USER and MARY will be created at the receiving workstation if they do not already exist.

NOTE: A fileset is always decompressed into the target directory that is specified when creating a package. In the above example, if the Create Directory Structure option is checked and the fileset is included in a package that has a default path of C:\SALES, then the INVOICE.DOC file will be decompressed into C:\SALES\USER\MARY.

11. Choose Save.

The changes made to a fileset are only committed to upon choosing the Save button. The Updating Fileset dialog box is displayed while the fileset contents are being saved and compressed. If you attempt to close the Edit Filesets dialog box without saving, you are prompted to save the fileset changes.

The fileset is created and added to the Filesets dialog box.

Managing Filesets

This section describes the procedures for editing, renaming, copying and deleting filesets.

Editing Filesets

Editing a fileset may be necessary in order to add or delete a file according to a change in a fileset's intent.

NOTE: It is recommended that you temporarily deactivate any packages which use the fileset you intend to edit.

Use the following procedure to edit the contents of a fileset.

1. Choose Tools | Filesets.

The Filesets dialog box is displayed.

2. Select the desired fileset and choose Edit.

A fileset can also be selected for editing by double clicking on the fileset name in the Filesets dialog box. The Edit Fileset dialog box is displayed listing all files included in the fileset.

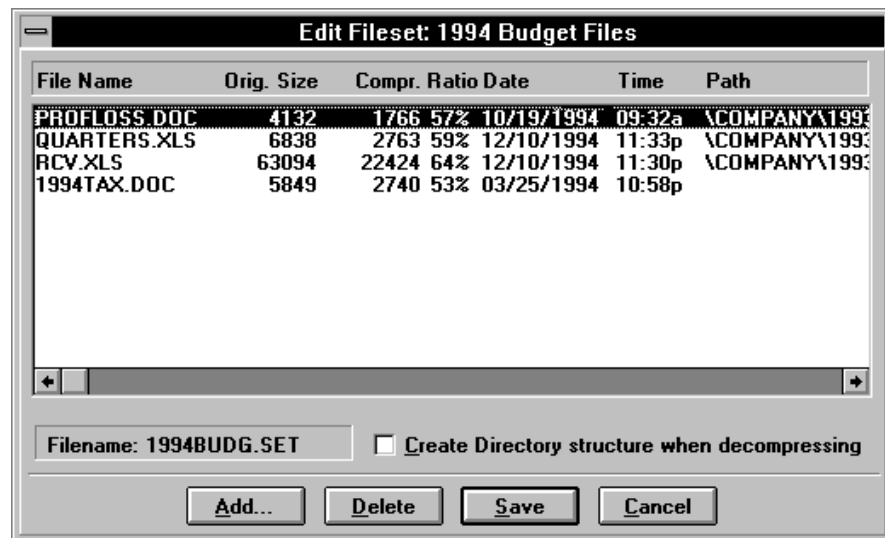


Figure 5-7: Editing a Fileset

For each file in the fileset, the following information displays:

Field	Description
File Name	the name of the file
Original Size	the file size before compression
Compressed Size	the file size after compression
Ratio	the compression ratio
Date	the file's creation date
Time	the file's creation time
Path	the file's path which displays only if the Include Path option is checked in the Add File dialog box

NOTE: Some files may show a 0% compression ratio. This occurs when the file is already compressed or when the file is very small.

- To add a file to the fileset, choose Add.
The Add File dialog box is displayed. Refer to the section above entitled "Creating Filesets" for detailed procedures on using this dialog box.
- To delete a file from the fileset, select the desired file name and choose Delete.
A prompt displays asking you to confirm the deletion. Choose Yes to continue with the delete action.

If deleted, the file name is removed from the Edit Filesets dialog box.

5. Choose Save.

The changes made to a fileset are only committed to upon choosing the Save button. The Updating Fileset dialog box is displayed while the fileset contents are being saved and compressed. If you attempt to close the Edit Filesets dialog box without saving, you are prompted to save the fileset changes.

Renaming Filesets

Changing the name of an existing fileset renames all instances of the former fileset name. For example, the new fileset name is reflected in the Filesets dialog box as well as in any packages which include the fileset. Use the following procedure to rename a fileset.

NOTE: A fileset can be renamed even if it is part of an actively scheduled package. The name of the .SET file which is maintaining the compressed fileset and is stored in the fileset directory does not change.

1. Choose Tools | Filesets.

The Filesets dialog box is displayed.

2. Select the desired fileset name and choose Rename.

The Rename Fileset dialog box is displayed prompting you to enter a new fileset name.

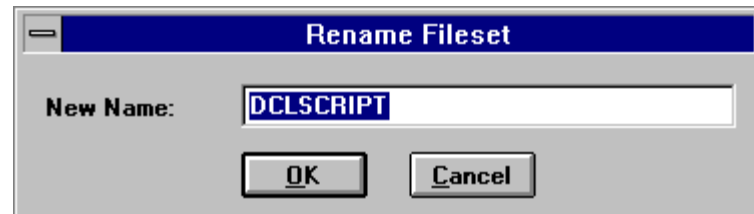


Figure 5-8: Renaming Filesets

3. Enter the new fileset name.

The new fileset name displays in the Filesets dialog box, and the old name is removed. All attributes of the old fileset are preserved in the renamed fileset (i.e., the fileset contents do not change).

4. Choose OK

Copying Filesets

NOTE: A fileset can be copied even if the original fileset is part of an actively scheduled package.

Use the following procedure to copy a fileset.

1. Choose Tools | Filesets.

The Filesets dialog box is displayed.

2. Select the desired fileset name and choose Copy.

The Copy Fileset dialog box is displayed prompting you to enter a new fileset name.

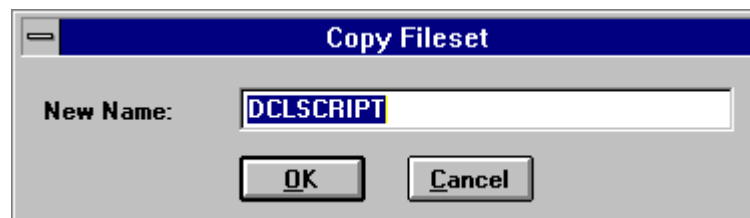


Figure 5-9: Copying Filesets

3. Enter the new fileset name,.

The new fileset is added to the Filesets dialog box. The new fileset contents are identical to the original fileset contents.

4. Choose OK

Deleting Filesets

NOTE: A fileset that is part of a scheduled package cannot be deleted.

Use the following procedure to delete a fileset.

1. Choose Tools | Filesets.

The Filesets dialog box is displayed.

2. Select the desired fileset name and choose Delete.

A prompt displays asking you to confirm the deletion.

2. Choose Yes to delete the fileset.

The fileset name is removed from the Filesets dialog box.

Chapter 6 *Scripts*

Chapter 5 discussed creating and managing filesets. This chapter discusses creating and managing scripts—a series of commands to be executed on a remote workstation.

Introduction

BrightWorks offers two different scripting languages: QuickScript and Powerscript. Scripts must be written according to a defined syntax, and they must be compiled successfully to be included in a package.

NOTES: a - The commands and instructions for using BrightWorks' QuickScript software distribution scripting language are documented in Chapter 7 of this manual. BrightWorks is shipped with several script files that can be customized for your own use. Refer to the section on page 74 entitled "Incorporating BrightWorks Scripts."

b - For further information regarding BrightWorks' PowerScript, refer to your PowerScript (DCL) manual.

Script Features

Sending scripts from a central location can maintain network workstation consistency and standardization. With scripts, you can easily do the following:

- update system executables and/or drivers (e.g., operating systems, network drivers)
- update system files (e.g., AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, network login script)
- install software on a user's workstation

A user can create a new script, as well as edit, compile, copy, rename and delete a script. The steps for each procedure are discussed in this chapter.

Creating Scripts

A script is created by assigning both a script name and a file name to the new script. The script *name* is used for identification purposes within BrightWorks. For example, it is immediately obvious that the script named “Upgrade to Win 3.1” is responsible for upgrading the Windows software to version 3.1. The script *file name* identifies the ASCII text file containing the script commands. The script file name must be a valid DOS file name (e.g., 8 characters plus the 3 character extension). After assigning the script name and file name, an “empty” script is created. The empty script must be edited in order to add commands. Use the following procedure to create a new script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed, listing the script names, the last compilation date, the status and the file name for each defined script.

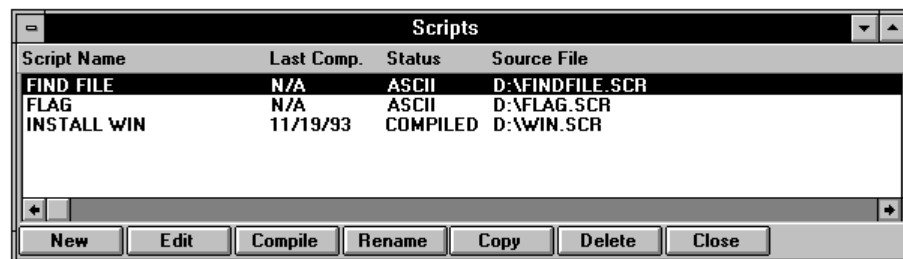


Figure 6-1: A List of Scripts

2. Choose New.

The Open New Script dialog box is displayed prompting you to enter the name, file name and destination directory for the new script.

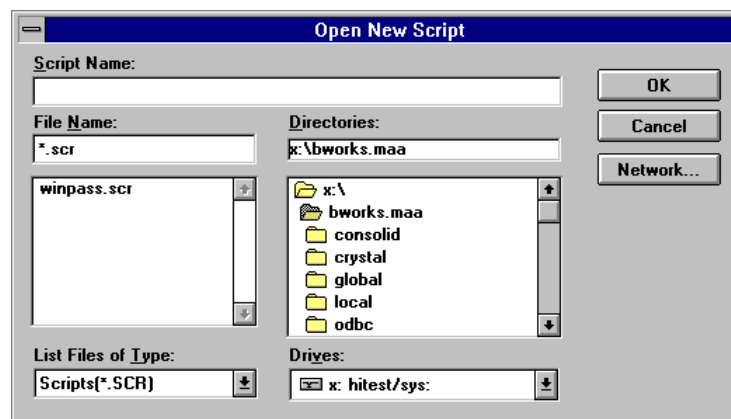


Figure 6-2: Creating a New Script

3. Enter the new script information.

The script *name* can be up to 80 characters, and all typed characters are valid. The script *file name* must follow the standard DOS conventions.

NOTES: a - Assigning all script files with the extension .SCR is recommended. A script file is a text file and can be edited with an external editor.

b - The scripts must reside on a network drive to which all users who will receive the script have access

4. Choose OK

The message “This file does not exist. Create the file?” displays.

5. Choose Yes to create the script file.

The Script Editor window is displayed.

The script name being edited displays in the title bar of the Script Editor window. All commands that are included in this script are listed. (The list is empty for new scripts.)

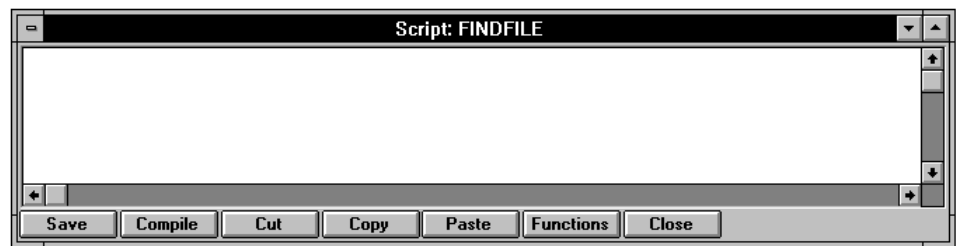


Figure 6-3: Entering Script Commands

6. Type the script commands.

Script commands can be typed directly into the Script Editor window. You can also select commands from a list by choosing Functions in the Script Editor window or the Edit | Paste Script Function (refer to the explanation below).

The script compiler requires one command per line. No error checking is performed until the script is compiled.

Optional comments can be placed in the script preceded by a semi-colon. These comments are ignored at compile time. For example:

```
;This is a comment.
```

NOTE: The commands and rules for using the scripting language are documented in Chapter 7, “Software Distribution Script Language.”

Standard editing functions are available from the Edit menu on the BrightWorks menu bar. The commands that are available from the Edit menu are listed in the following table.

Command	Description
Undo	Removes the last change made to the script.
Cut	Copies a block of selected text to the clipboard and removes the text from the Script Editor window.
Copy	Copies a block of selected text to the clipboard.
Paste	Places the block of text from the clipboard into the Script Editor window at the current cursor location.
Paste Script Function	Displays the Choose Script Function dialog box, as in Figure 6-4. This dialog box allows you to select a function (from a list of all script functions) to be placed in the script at the current cursor location. A function can be selected by either double clicking on the function name, or highlighting the name and choosing the OK button. Choosing the Help button displays help text for the highlighted function.
Find	Searches the script for a user-specified text string.
Next	Searches the script for the next occurrence of the user-specified text string.
Replace	Searches the script for a user-specified text string and replaces the found text with another user-specified text string.
Fonts	Enables you to select the font, style and size of the script type.

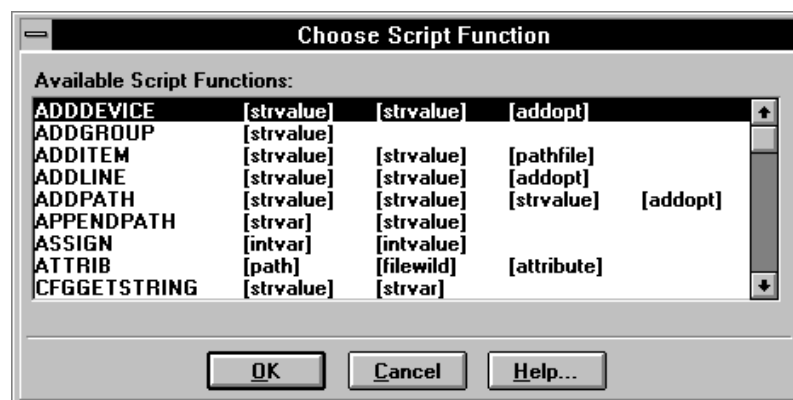


Figure 6-4: Choosing a Function to Place in the Script

NOTE: During script editing, the status bar in the BrightWorks application window indicates the current line and column position of the typing cursor.

7. To compile the script, choose Compile.

Refer to "Compiling Scripts" on page 69 for further information.

8. To save the script contents, choose Save.

The saved script contents are stored in ASCII text format. The script must be compiled to be used in a package. To compile the script, follow the procedure below entitled "Compiling Scripts."

9. Choose Close to close the Script Editor window.

If you did not save the script changes as in Step 6 above, you are prompted to do so now. Choose Yes to save the script changes, or choose No to close the Script Editor without saving any changes.

The new script is added to the Scripts window. The status of all uncompiled scripts is 'ASCII.' A script must be compiled to be used in a package.

Compiling Scripts

The Status field in the Scripts window indicates the status of each script. Script status can be either ASCII or COMPILED. A script's status must be COMPILED to be used in a package for distribution.

The commands and instructions for using the scripting language are documented in Chapter 7, "Software Distribution Script Language." The compilation process checks the syntax and validity of the script's commands.

Compiling Scripts

Use the following procedure to compile a script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

2. Select the desired script in the Scripts window and choose Compile.

While a compile is in progress, the Compile Status dialog box is displayed.

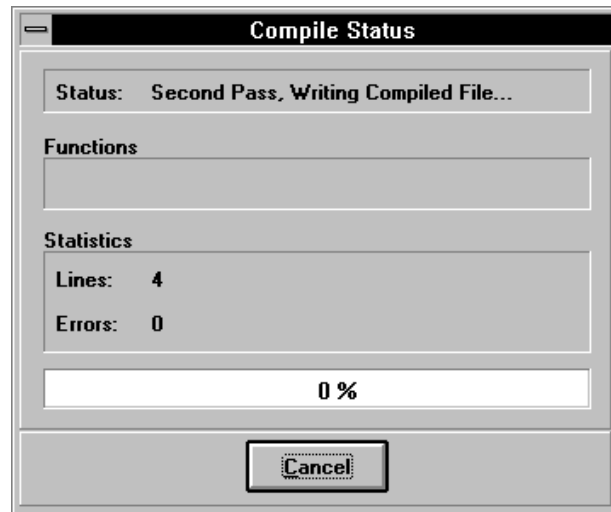


Figure 6-5: Compiling a Script

When the compile is complete, the Status field in the Compile Status dialog box indicates success or failure. If the compile fails, the Function field indicates the first function found which has invalid parameters. The Statistics area indicates the total number of lines in the script (Lines field) and the number of errors found (Errors field).

3. Choose OK to continue.

If the script compile is successful, then choose Close in the Script Editor window to return to the Scripts window which shows the script's status as COMPILED.

If the script compile fails, then the Compiler Messages dialog box is displayed listing the first script line which contains errors, as in Figure 6-6.



Figure 6-6: Message Indicating the Script Line in Error

4. To correct a compiler error condition, double click on an error line in the Compiler Messages dialog box.

The Script Editor window is displayed with the script that you are attempting to compile. The selected error line is automatically highlighted.

5. Correct all error conditions, and attempt to re-compile the script.

Refer to Chapter 7 for details on the scripting rules and commands.

After successful compilation of the script, the script can be used in a package.

NOTE: If you edit a script that has already been compiled, the script must be successfully re-compiled in order to be used in a package. Refer to the “Last Comp” field in the Scripts window to discover the date on which the file was last compiled.

Managing Scripts

Editing Scripts

Editing a script may be necessary for either of the following reasons:

- Existing scripts might need to be edited in order to add or delete commands according to a change in a script’s intent.
- When a script compilation fails, the script must be edited to resolve the error(s).

NOTE: Temporarily deactivating any packages which use the script you intend to edit is recommended.

Use the following procedure to edit the contents of a script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

2. Select the desired script and choose Edit.

A script can also be selected for edit by double clicking on the script name in the Scripts window. The Script Editor window is displayed.

The script name being edited displays in the title bar of the Script Editor window. All commands that are included in this script are listed.

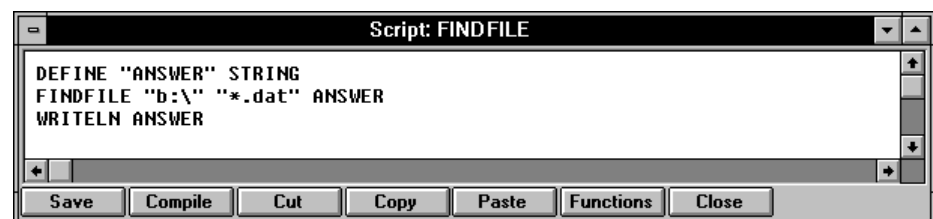


Figure 6-7: Entering Script Commands

3. Edit the script commands.

Script commands can be directly typed into the Script Editor window. Commands can also be selected from a list of commands by choosing the Functions button in the Script Editor window or the Edit | Paste Script Function.

The script compiler requires one command per line. No error checking is performed until the script is compiled.

NOTES: a - The commands and rules for using the scripting language are documented in Chapter 7, "Software Distribution Script Language."

b - During script editing, the status bar in the BrightWorks application window indicates the current line and column position of the typing cursor.

4. To compile the script, choose Compile.

Refer to "Compiling Scripts" on page 69 for further information.

5. To save the edited script contents, choose Save.

6. Choose Close to exit the Script Editor window.

NOTE: If you edit a script that has already been compiled, the script must be successfully re-compiled in order to be used in a package.

Renaming Scripts

Changing the name of an existing script renames all instances of the former script name. For example, the new script name will be reflected in the Scripts window as well as in any packages which include the script.

NOTE: A script can be renamed even if it is part of an actively scheduled package.

Use the following procedure to rename a script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

2. Select the desired script and choose Rename.

The Rename Script dialog box is displayed prompting you to enter a new script name.

3. Enter the new script name.

The new script name displays in the Scripts window, and the old name is removed. All attributes of the old script are preserved in the renamed script (i.e., the script contents do not change).

4. Choose OK.

NOTE: The script rename procedure only changes the script name—the script file name does not change.

Copying Scripts

NOTE: A script can be copied even if the original script is part of an actively scheduled package.

Use the following procedure to copy a script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

2. Select the desired script and choose Copy.

The Copy Script dialog box is displayed prompting you to specify a name, file name and destination directory for the new script. The script name can be up to 80 characters, and all typed characters are valid. The script file name must follow the standard DOS conventions and can reside in any directory.

NOTE: Assigning all script files with the extension .SCR is recommended.

3. Enter the new script information.

The new script name is added to the Scripts window. The new script is populated with the same commands as the original script.

4. Choose OK.

Deleting Scripts

NOTE: A script that is part of a scheduled package cannot be deleted.

Use the following procedure to delete a script.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

2. Select the desired script and choose Delete.

A prompt displays asking you to confirm the deletion.

3. Choose Yes to delete the script.

If deleted, the script name is removed from the Scripts window.

NOTE: The delete action only deletes the script name from the Scripts window. The corresponding .SCR file is not deleted. Therefore, if a script name is inadvertently deleted, you can create a new script and assign the same script file name to retrieve the deleted script contents.

Incorporating BrightWorks Scripts

BrightWorks is shipped with several pre-defined scripts that can be customized for use in your environment. Upon BrightWorks installation, the script files are copied into the BWORKS.200 program directory.

The table below lists the purpose of each script and indicates the script file name:

Purpose	File Name
Word for Windows Local Installation	LOCAL.SCR
Find and Delete a Program File	FINDDEL.SCR
AUTOEXEC.BAT Replacement	AUTORLP.SCR
AUTOEXEC.BAT Append	AUTOAPP.SCR
AUTOEXEC.BAT Modification	AUTOMOD.SCR
CONFIG.SYS Replacement	CFGRPLMT.SCR
CONFIG.SYS Append	CFGAPP.SCR
CONFIG.SYS Modification	CFGMOD.SCR
DOS Upgrade 3.X to 6.X	DOS3TO6.SCR
DOS Upgrade 4.X to 5.X	DOS4TO5.SCR
DOS Upgrade 5.X to 6.X	DOS5TO6.SCR
Novell NETX Update	NETXUP.SCR
Novell Wkst ODI/VLM/IPX Driver Startup Batch File Update	NETBAT.SCR
VLM Upgrade	VLMUPGRD.SCR
Send Text Message to Network Users	TYPE.SCR
Copy Files to Server	CPFS.SCR
Windows Network Installation	WININST.SCR
Windows Add Program Group	ADDGROUP.SCR
Windows Add Program Item	ADDITEM.SCR

Windows INI Replacement	INIRPL.SCR
Windows INI Append	WINIAPPD.SCR
Windows INI Modification	WININIMD.SCR
Windows Spooler Setting in INI	WINSPOOL.SCR
Windows Wallpaper Update	WALLPAPR.SCR
CC:MAIL for Windows installation	NETINS.SCR

Use the following procedure to incorporate a pre-defined script into BrightWorks.

1. Choose Tools | QuickScripts.

The Scripts window is displayed.

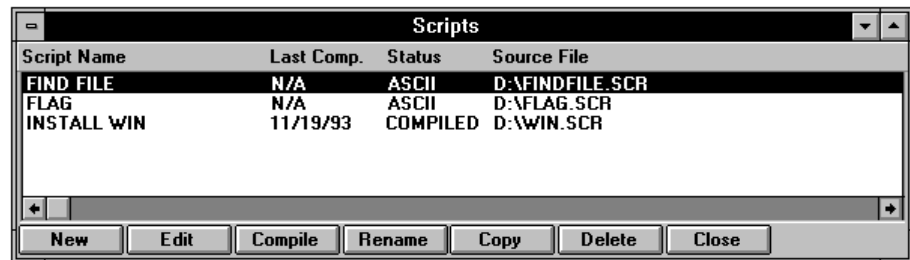


Figure 6-8: A List of Scripts

2. Choose New.

The Open New Script dialog box is displayed.

3. Enter a name for the script in the Script Name field.

The script name can be up to 80 characters, and all typed characters are valid. The script name is used within BrightWorks to identify the script. For example, if you are incorporating the script which adds a group to the Program Manager desktop, then you might want to define the script name as ADD PROGRAM GROUP.

4. Select the script to be incorporated into BrightWorks.

Select a script from the list of script file names. For example, if you want to incorporate and edit the script which adds a group to the Program Manager desktop, then you would select the ADDGROUP.SCR file.

After entering a script name and selecting a script filename, the Open New Script dialog box should look similar to Figure 6-9.

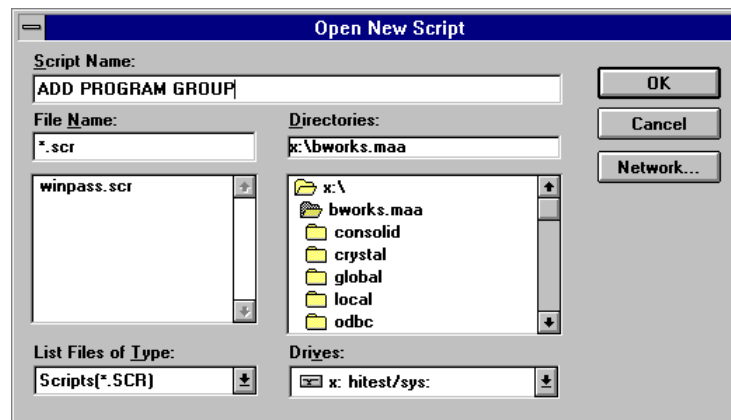


Figure 6-9: Incorporating a Pre-defined Script

5. Choose OK.

The Script Editor window is displayed listing the commands for the chosen script.

6. Edit the script commands.

The commands and rules for using the scripting language are documented in Chapter 7, "Software Distribution Script Language."

7. Choose Compile.

Refer to "Compiling Scripts" on page 69 for further information.

8. Choose Save.

The script contents are saved.

9. Choose Close to exit the Script Editor window.

The new script is added to the Scripts window.

NOTE: A script must be compiled to be used in a package.

Chapter 7 Software Distribution QuickScript Language

Chapter 6 discussed creating and managing scripts. This chapter lists the variables and rules for each function in the BrightWorks QuickScript scripting language.

For detailed information regarding BrightWorks' PowerScript, refer to your PowerScript (DCL) manual.

Introduction

A script is a series of commands to be executed on a remote workstation. Scripts must be written according to a defined syntax, and they must be compiled successfully to be included in a package.

The commands and instructions for using the BrightWorks QuickScript language are discussed in this chapter. The procedures for creating, compiling and managing scripts are discussed in the Chapter 6, "Scripts."

Notes on Syntax

The following items must be noted when writing scripts:

- Only one command can be placed on a line.
- The syntax for each command/function is as follows:
- FUNC_NAME [parameter1], [parameter2], ...[parameterN]
- Unless otherwise noted, each function returns a 0 if it is successful (i.e., the system variable [retval] is set to 0). The action to be taken as a result of a script's return code is defined when the script is included in a package. These "Advanced Package Options" are discussed in "Creating and Editing Packages" in Chapter 9, "Packages."
- Some functions take "optional" parameters. The administrator should decide whether or not to specify these parameters. If they are not specified, an empty or NULL value must be passed to the compiler to act as a placeholder.

For example, the COPY function has the following parameters:

```
COPY [path] [filewild] [path] {filewild}
```

where the last parameter, {filewild}, is optional. The COPY command below provides an example for copying all .BAT files from the C: drive to the B: drive, using a placeholder to stand for the last {filewild} parameter:

```
COPY "C:\ " "*.BAT" "B:\ " "
```

In this example, the files are not renamed and retain their original .BAT extensions.

Script Functions

Function Set Overview

Each script “command” is treated as a “function” (e.g., a C function) that has two basic properties:

- each command has 0 to 4 parameters that it will be passed
- most commands have a return code

As such, the script language supports user defined variables (of integer and string type), as well as “system” variables. When necessary, the functions also implement return values from the parameters that are passed.

Each function has one or more parameters that can be passed. In the following discussions, the required parameters are surrounded by [], and the optional parameters are surrounded by { }. Each parameter is the name of a rule, whose allowable values are listed in “DOS Error Codes” on page 107 of this chapter.

Type of Functions

The script functions are divided into the four major categories summarized below:

DOS Functions

NOTE: Refer to “DOS Functions” on page 82 for detailed information about these functions.

Function	Required Parameters	Description
ATTRIB	[path] [filewild] [attribute]	Changes the attributes of a file or multiple files.
COPY	[path] [filewild] [path] {filewild}	Copies a file or files to another directory and file name(s).
DELETEDIR	[path] [filename] {deleteopt}	Deletes a directory.
DELETEFILE	[path] [filewild]	Deletes a file or multiple files.
FINDFILE	[path] [filewild] [strvar]	Finds a file.
MDIR	[path] [filename]	Creates a directory.
RENAME	[path] [filewild] [path] [filewild]	Renames a source file(s).
UPGRADEOS	[upgopt]	Upgrades DOS version from 3.x-5.x to either 5.00 or 6.00.

System File Functions

NOTE: Refer to "System File Functions" on page 88 for further information on system file function.

Function	Required Parameters	Description
ADDDEVICE	[strvalue1] [strvalue2] [addopt]	Adds a new DEVICE= line to a system file.
ADDLINE	[strvalue1] [strvalue2] [addopt]	Adds a line of text to a system file.
ADDPATH	[strvalue1] [strvalue2] [strvalue3] [addopt]	Adds a sub-directory to a path environment variable.
CFGGETVALUE	[strvalue] [intvar]	Retrieves a numeric variable from a system file.
CFGSETVALUE	[strvalue] [intvalue]	Sets a numeric variable in a system file.
CFGRETRIEVE STRING	[strvalue] [strvar]	Retrieves a string variable from a system file.

CFGSETSTRIN G	[strvalue1] [strvalue2]	Sets a string variable in a system file.
REPLACEKEY	[strvalue1] [strvalue2] [strvalue3]	Replaces a key value in a system file.
REPLACELINE	[strvalue1] [strvalue2]	Replaces an existing line in a system file with a new line.
REPLACELINE- ADD	[strvalue1] [strvalue2] [addopt]	Replaces or adds an existing line in a system file.
SETSYSFILE	[path] [filename]	Sets a system file to be manipulated.

Windows System File Functions

NOTE: Refer to "Windows System File Functions" on page 95 for further information on Windows System File Functions.

Function	Required Parameters	Description
ADDGROUP	[strvalue]	Creates a new Program Manager group.
ADDITEM	[strvalue1] [strvalue2] [pathfile]	Adds a new item to a Program Manager group.
GETINIINT	[pathfile] [strvalue1] [strvalue2] [intvar]	Retrieves a key value (integer) from an INI file, and places the result in a variable.
GETINISTR	[pathfile] [strvalue1] [strvalue2] [strvar]	Retrieves a key value (string) from an INI file, and places the result in a variable.
SCHEDULEWIN	[path] [filename] [text]	Schedules a file to be run the next time the user runs Windows.
WRITEINIINT	[pathfile] [strvalue1] [strvalue2] [intvalue]	Writes a key value (integer) to an INI file
WRITEINISTR	[pathfile] [strvalue1] [strvalue2] [strvalue3]	Writes a key value (string) to an INI file.

Miscellaneous Functions

NOTE: Refer to "Miscellaneous Functions" on page 100 for further information.

Function	Required Parameters	Description
APPENDPATH	[strvar] [strvalue]	Appends the contents of a string value to the end of a string variable; however, it first checks if the last character of a string variable is a "\". If it is not, APPENDPATH will append a "\" and a value to the variable.
ASSIGN	[intvar] [intvalue]	Performs a basic integer assignment operation.
DEFINE	[text] [defineopt]	Used to create user defined variables of a string or integer type.
EXIT	[intvalue]	Ends the script.
IF...THEN...ELSE	[intvalue1] [condoper] [intvalue2]	Allows conditional processing of functions.
NUMTOSTR	[strvar] [intvalue]	Converts a numeric value to a string variable.
PAUSE	[text]	Pauses execution of the script until the user presses a key.
REBOOT		Immediately reboots the user's PC.
SHELL	[pathfile] {text} {shellopt}	Allows a user to execute an external DOS batch file or executable program. Used to run DOS based PowerScripts.
STRCAT	[strvar] [strvalue]	Appends the contents of a string value to the end of a string variable.
STRCOMPARE	[strvar] [strvalue]	Does a byte for byte comparison of two strings.
STRCOPY	[strvar] [strvalue]	Copies a value into a string, overwriting the previous contents of the string.

WRITELN	[strvalue]	Writes a string value (e.g., write to screen).
---------	------------	--

NOTE: In the actual script, parameters are separated by a space—do not type the brackets.

User-defined Variables

User-defined variables are defined by using the DEFINE command (see Miscellaneous Functions) to create a string or integer user-defined variable name.

User-defined variables must be defined before listing any script functions. Also, the appropriate type must be used when calling a function that allows user-defined variable names. The functions that allow user-defined variables, system variables and literal text use the phrases [strvar] or [intvar] in their parameter listings. For further information, refer to "DOS Error Codes" on page 107.

DOS Functions

The DOS function set is used for managing a machine's files and directories. For example, files can be searched for, copied, deleted, renamed and tagged with a specified attribute; directories can be created and deleted.

Return values are generated when appropriate (unless otherwise noted, the functions return 0 if successful). Any applicable system variables are also noted.

Most DOS functions return a DOS error code if unsuccessful. Refer to the table on page 110 for a list of the DOS error codes that may be returned.

NOTES: a - When an "explicit <path>" is mentioned, it can take the form of D:\PATH (SERVER/VOLUME:\PATH is not currently supported).

b - Some functions take optional "options." The administrator should decide whether or not to specify these options.

c - In the following function specifications, parameters in quotes represent literal parameters; all other parameters represent rules. Refer to "Rules and System Variables" on page 107 for a full listing of rules.

ATTRIB [path] [filewild] [attribute]

Parameter	Description and Notes
[path]	Source path to files. This path must exist.
[filewild]	The file name whose attributes are to be changed. May contain wildcards (? and *).
[attribute]	RO - Read only RW - Read/Write A - Set Archive bit SY - System file H - Hidden file SH - Shareable (network <path> only) -A - Turn off archive attribute -SY - Turn off system attribute -H - Turn off hidden attribute -SH - Turn off shareable attribute (network <path> only)

Description - Changes the attributes of a file or multiple files.

Tip - To remove the Read Only attribute, use the RW attribute. (There is no -RO attribute.)

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = -1 if the SH or -SH attributes are used and the drive letter specified in [PATH] is not a network drive

[RETVAL] = -2 if the SH or -SH attributes are used and no drive letter is specified in [PATH]

[RETVAL] = DOS error code in all other cases

Example - Set the AUTOEXEC.BAT file on a user's boot drive to Read Only:

```
ATTRIB [BOOT_ROOT] "AUTOEXEC.BAT" RO
```

COPY [path] [filewild] [path] {filewild}

Parameter	Description and Notes
[path]	Source path of file to be copied.
[filewild]	Source file name to be copied. May contain standard DOS wild cards (? and *).
[path]	Destination path.
{filewild}	<i>Optional</i> destination file name. (If not specified, *.* is assumed.) May contain standard DOS wild cards (? and *). May be used to rename file(s) during file copy. If not used, the placeholder "" or NULL must be specified.

Description - Copies a file or files to another directory and file name(s).

Return Values:

[RETVAL] = 0 if file(s) are copied correctly

[RETVAL] = DOS error code if the function is unsuccessful

Example - Copy the WIN.INI file from the Windows directory found at login to the local Windows directory. Two examples of this are:

```
COPY [WINDIR] "WIN.INI" "C:\WINDOWS" ""
```

or

```
COPY [WINDIR] "WIN.INI" "C:\WINDOWS" NULL
```

DELETEDIR [path] [filename] {deleteopt}

Parameter	Description and Notes
[path]	Source path to the directory to be deleted. This path must exist.
[filename]	Directory name to be deleted.
{deleteopt}	<i>Optional</i> delete option: ALL - causes DELETEDIR to delete the specified directory and everything under it, including any subdirectories, hidden, system and read only files. If not used, NULL must be specified.

Description - Deletes a directory.

Tip - Use the ALL delete option with caution since it can delete entire directory trees.

Return Values:

[RETVAL] = 0 if the directory is successfully deleted

[RETVAL] = DOS error code if the function is unsuccessful

Example - Delete the Windows directory found at login and all of its files and sub-directories:

```
DELETEDIR [WINDIR] ALL
```

DELETEFILE [path] [filewild]

Parameter	Description and Notes
[path]	Source path to the file(s) to be deleted. This path must exist.
[filewild]	File name(s) to be deleted. Wild cards may be specified (? and *) to delete multiple files.

Description - Deletes a file or multiple files.

Return Values:

[RETVAL] = 0 if the file(s) are deleted

[RETVAL] = DOS error code if the function is unsuccessful

Example - Delete all .DOC files from the F:\UZR\JOHN sub-directory:

```
DELETEFILE "F:\UZR\JOHN" "*" .DOC "
```

FINDFILE [path] [filewild] [strvar]

Parameter	Description and Notes
[path]	Source path in which to search for files. This path must exist.
[filewild]	The search criteria. May contain wildcards (? and *).

[strvar]	A string variable which contains the file name of the first file found. (Before being used as a parameter, this variable must be defined using the DEFINE function.)
----------	---

Description - Finds a file.

Return Values:

[RETVAL] = 0 and copies the name of the first file found into [STRVAL] if successful

[RETVAL] = -1 and sets [STRVAL] to NULL if no files are found

Example - Test for the presence of the NET.CFG file in the [NET.CFG] directory:

```
DEFINE "Result" STRING
FINDFILE [NETCFG] "NET.CFG" RESULT
```

MDIR [path] [filename]

Parameter	Description and Notes
[path]	Path in which to create the new directory. This path must exist.
[filename]	Sub-directory to create. Wild cards may not be specified.

Description - Creates a directory.

Return Values:

[RETVAL] = 0 if the directory is successfully created

[RETVAL] = DOS error code if the function is unsuccessful

Example - Create the JOHN sub-directory in the UZR directory:

```
MDIR "F:\UZR" "JOHN"
```

RENAME [path] [filewild] [path] [filewild]

Parameter	Description and Notes
[path]	Source path to files to be renamed. This path must exist.

[filewild]	Source file name to be renamed. May contain wildcards (? and *).
[path]	Destination path (can be different than [path] to enable moving files, but the drives must be the same).
[filewild]	New file name. May contain wildcards (? and *). If so, the standards used by the DOS REN command are followed.

Description - Renames a source file(s).

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Rename all .BAT files in the C:\ drive to .BAK:

```
RENAME "C:\ " "*.BAT" "C:\ " "*.BAK"
```

UPGRADEOS [upgopt]

Parameter	Description and Notes
[upgopt]	5.00 - upgrade DOS version to 5.00 6.00 - upgrade DOS version to 6.00

Description - Upgrades DOS version from 3.x-5x to either 5.00 or 6.00.

NOTE: Do not use this function on a workstation that has Windows NT installed in a dual boot configuration. It will cause the boot menu to be lost. The PC will boot DOS only.

Tips:

1) In order for BrightWorks to have access to the upgraded DOS files, EQUIP must first be run on a machine that has the desired DOS files. For example, to upgrade a machine's DOS version to 6.00, EQUIP first must be executed on a machine that has DOS 6.00. By executing EQUIP from the same directory in which the BrightWorks software distribution update program (SDUPDATE.EXE) file is located, the DOS files become accessible by BrightWorks. Further, the machine on which EQUIP is run must not contain any system that modifies the machine's boot record (e.g., OS/2, Windows NT).

2) The machine must be rebooted after the script is executed. Use the REBOOT function as the last script function.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Upgrade a user's DOS version to 5.00:

```
UPGRADEOS 5.00
IF [RETVAL]=0
    ...
    ; copy DOS files, edit CONFIG.SYS, etc.
    REBOOT
ENDIF
```

System File Functions

The System File functions allow for easy manipulation of basic system files, such as CONFIG.SYS, AUTOEXEC.BAT, NET.CFG, or a login script. (Use the Windows System File functions to edit .INI files.)

Most System File functions return a DOS error code if unsuccessful. Refer to the table on page 110 for a listing of the DOS error codes.

NOTE: Prior to using any of the functions in this category, you must call SETSYSFILE. Also, none of the functions will create a backup of the file that they are modifying; however, a file will not be modified if a function fails. It is your responsibility to backup any files as necessary.

All of the System File functions make use of a “key” value. This value is used to search the file to aid in determining where to make a modification. All key searches are *case insensitive*. If a key is found, its corresponding value is defined as the first non- whitespace (e.g. tab, cr/lf, =, etc.) group of characters after the found key value. For example, consider the following line:

```
PATH=C:\DOS;C:\WINDOWS
```

If “PATH” is specified as the key, then the corresponding value is “C:\DOS;C:\WINDOWS.” However, consider the following line:

```
STACKS 9,256
```

If “STACKS” is specified as the key, then the corresponding value is “9,256.” As a result, an equal sign is not necessary to identify a value that you might want to edit.

NOTE: In the following function specifications, parameters in quotes represent literal parameters; all other parameters represent rules. Refer to "Rules and System Variables" on page 107 for a full listing of rules.

ADDDEVICE [strvalue1] [strvalue2] [adopt]

Parameter	Description and Notes
[strvalue1]	The path and driver name (e.g. C:\WINDOWS\EMM386.EXE).
[strvalue2]	The key value to search for (e.g. HIMEM.SYS).
[adopt]	Where [strvalue1] is to be placed: either BEFORE or AFTER [strvalue2].

Description - Adds a new DEVICE= line to a system file (typically the DOS CONFIG.SYS).

Tip - If [strvalue2] is a null string or the key is not found, ADDDEVICE will add [strvalue1] in the position of the file indicated by [adopt].

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Place "DEVICE=C:\WINDOWS\EMM386.EXE" after the "DEVICE=HIMEM.SYS" line in the CONFIG.SYS file:

```
SETSYSFILE "C:\ " "CONFIG.SYS"
ADDDEVICE "C:\WINDOWS\EMM386.EXE" "HIMEM.SYS" AFTER
```

ADDLINE [strvalue1] [strvalue2] [adopt]

Parameter	Description and Notes
[strvalue1]	The entire line of text you want to add.
[strvalue2]	A reference key value to be positioned relative to [strvalue1]. This is a "keyword" that will be searched for in the file. Specify as much or as little as you like. When the first occurrence of the keyword in a line is found, that line is used as the reference.
[adopt]	Specify where [strvalue1] is to be placed: either BEFORE or AFTER [strvalue2].

Description - Adds a line of text to a system file.

Tip - If [strvalue2] is a null string, ADDLINE will place [strvalue1] in the position of the file indicated by [addopt].

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Add a new line to the end of a user's CONFIG.SYS file:

```
SETSYSFILE "C:\ " "CONFIG.SYS"
ADDLINE "THIS IS NEW.." " " AFTER
```

NOTE: As in the example above, non-specified parameters (e.g., [strvalue2]) can be indicated by empty quotes. Entering NULL with no quotes is also acceptable.

ADDPATH [strvalue1] [strvalue2] [strvalue3] [addopt]

Parameter	Description and Notes
[strvalue1]	The name of the path environment variable to edit (PATH for DOS, or DPATH for OS/2, or TEMP, etc.).
[strvalue2]	The sub-directory to be added.
[strvalue3]	The sub-directory that [strvalue2] will be placed either before or after.
[addopt]	Specify where [strvalue2] is to be placed: either BEFORE or AFTER [strvalue3].

Description - Adds a sub-directory to a path environment variable.

Tips:

- 1) If [strvalue3] is a null string, ADDPATH will place [strvalue2] in the position of the path statement indicated by [addopt] (i.e., the new path will be placed at the beginning or end of the path statement).
- 2) If the key specified in [strvalue1] is not found, then a new one is added, with a "SET" prepended. This allows for adding path like environment variables such as "SET TEMP=", and so on.
- 3) This function can also be used to edit other lines such as a TEMP environment variable, or any other line that does something like "SET envvar=d:\path."

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Add the sub-directory WINDOWS to the path and place it before the DOS variable in the AUTOEXEC.BAT file:

```
SETSYSFILE "C:\\" "AUTOEXEC.BAT"
ADDPATH "PATH" "C:\WINDOWS" "C:\DOS" BEFORE
```

CFGGETVALUE [strvalue] [intvar]

Parameter	Description and Notes
[strvalue]	The variable to be retrieved.
[intvar]	An integer variable to hold the retrieved value. (Before being used as a parameter, this variable must be defined using the DEFINE function.)

Description - Retrieves a numeric variable from a system file (e.g., FILES, BUFFERS, etc.).

Tip - If the value of the key specified is non-numeric (e.g., the DOS=HIGH), CFGGETVALUE sets parameter 2 to 0, but does not return an error code. Use CFGRETRIEVESTRING to get a string value.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = -2 if the key value could not be found

[RETVAL] = DOS error code in all other cases

Example - Place the value of the FILES= statement in the CONFIG.SYS file into a user defined variable called nRESULT (which must first be defined!):

```
DEFINE "nRESULT" STRING
SETSYSFILE "C:\\" "CONFIG.SYS"
CFGGETVALUE "FILES" nRESULT
```

CFGSETVALUE [strvalue] [intvalue]

Parameter	Description and Notes
[strvalue]	The variable to be set.

[intvalue]	The integer value.
------------	--------------------

Description - Sets a numeric variable in a system file (e.g., FILES, BUFFERS, etc.).

Tip: Use ADDLINE to add a new statement if one does not exist.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = -2 if the key value could not be found

[RETVAL] = DOS error code in all other cases

Example - Set the value of the FILES= statement in the CONFIG.SYS file to 50, provided a FILES= statement already exists in the file:

```
SETSYSFILE "C:\ " "CONFIG.SYS"
CFGSETVALUE "FILES" 50
```

CFGRETRIEVESTRING [strvalue] [strvar]

CFGSETSTRING [strvalue1] [strvalue2]

These two functions act exactly the same as CFG???VALUE, except they deal with string values rather than integer values. An administrator might use this to check non-numeric variables (e.g., STACKS=9,256 is a non numeric value).

Note that before using the [strvar] variable as a parameter, the variable must be defined using the DEFINE function.

REPLACEKEY [strvalue1] [strvalue2] [strvalue3]

Parameter	Description and Notes
[strvalue1]	The line in the system file which contains the key value to be replaced.
[strvalue2]	The key value to be replaced.
[strvalue3]	The new value.

Description - Similar to REPLACELINE; however, it replaces a key value rather than the entire line.

Tip - If [strvalue3] is a null string, [strvalue2] will be removed.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Change the “40” to a “50” in the FILES= line in the CONFIG.SYS file:

```
SETSYSFILE "C:\\" "CONFIG.SYS"
REPLACEKEY "FILES=40" "40" "50"
```

REPLACELINE [strvalue1] [strvalue2]

Parameter	Description and Notes
[strvalue1]	The key value of the line you wish to replace, such as PATH, COMSPEC or DEVICE.
[strvalue2]	The new value of the entire line.

Description - Replaces an existing line in a system file with a new line.

Tips:

- 1) If [strvalue2] is a null string, then the line will be deleted.
- 2) If the key value exists more than one time in the file, only the first instance is modified.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Replace the existing COMSPEC line in the CONFIG.SYS file with a new line:

```
SETSYSFILE "C:\\" "CONFIG.SYS"
REPLACELINE "COMSPEC" "SET COMSPEC=C:\DRDOS\COMMAND.COM"
```

REPLACELINEADD [strvalue1] [strvalue2] [adopt]

Parameter	Description and Notes
[strvalue1]	The key value of the line you wish to replace, such as PATH, COMSPEC or DEVICE.
[strvalue2]	The new value of the entire line.
[adopt]	Where [strvalue1] is to be placed: either BEFORE or AFTER [strvalue2].

Description - Similar to REPLACELINE, this function replaces an existing line in a system file with a new line. However, if the key specified in [strvalue1] is not found, then the line specified in [strvalue2] is added to the file, at the beginning or end of the file depending on the position defined by [addopt].

Tip: If [strvalue1] is not found, then the line specified as [strvalue2] will be added to the file in the position defined by [addopt].

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Replace the existing NETX line with the new line C:\NET\VLM. If NETX is not found, then the line will be appended to the end of the file:

```
SETSYSFILE "C:\\" "NET.BAT"
REPLACELINEADD "NETX" "C:\NET\VLM" AFTER
```

SETSYSFILE [path] [filename]

Parameter	Description and Notes
[path]	The path to the file to be modified.
[filename]	The name of the file to be modified.

Description - Sets a system file to be manipulated.

Tips:

1) This function must be called prior to calling any of the functions in the Easy System File function category. It needs to be called only once, unless you change the file you are working on in the script.

2) Using [BOOT_ROOT] as the [path] parameter will *always* modify the file on the boot disk, regardless of whether or not the user is given the option to override the installation path (in the package definition). Use [TARGET] as the [path] parameter if the user is given the option to override the installation path.

Return Values:

[RETVAL] = 0 if file is found

[RETVAL] = 2 if file is not found

Example - Designate a user's CONFIG.SYS file as the file to be edited. Two examples of this are:

```
SETSYSFILE "C:\\" "CONFIG.SYS"
      or
SETSYSFILE [BOOT_ROOT] "CONFIG.SYS"
```

Windows System File Functions

The Windows System File functions provide the ability to edit INI files and create and manipulate Program Manager groups.

In the following function specifications, parameters in quotes represent literal parameters; all other parameters represent rules. Refer to "Rules and System Variables" on page 107 for a listing of the rules.

Many of the Windows System File functions have a [pathfile] parameter which specifies the path name and file name to an INI file. If you do not specify a full path to the Windows directory, then the actions performed by these functions occur on the first instance of Windows found, as determined by the path statement of the receiving machine. If Windows is not found in the path, then the distribution update program will search for the INI file in [BOOT_ROOT]\WINDOWS. If Windows is still not found, the update program will then try [BOOT_ROOT]\WIN31.

The functions ADDGROUP, ADDITEM and SCHEDULEWIN use the WSDUPD.EXE update program which is copied into the local Windows directory each time these functions are used. The next time the user runs Windows, WSDUPD.EXE runs and executes the appropriate function(s). It then deletes WSDUPD.EXE and WSDUPD.INI. If a user has SHARE.EXE loaded, a "sharing violation" message will display when trying to delete WSDUPD.EXE. This message can be ignored.

ADDGROUP [strvalue]

Parameter	Description and Notes
[strvalue]	The string which specifies the name of the Program Manager group to be added.

Description - Creates a new Program Manager group.

Tip: When the ADDGROUP script function is executed, the BrightWorks software distribution update program WSDUPD.EXE is automatically copied into the workstation's Windows directory. The WSDUPD.EXE command is also added to the "Load=" line in the WIN.INI file. The next time Windows is run at the workstation, the function is executed and WSDUPD.EXE is removed from the WIN.INI "Load=" line.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code if unsuccessful. The function might fail if WSDUPD.EXE could not be copied into the Windows directory or if the WSDUPD.EXE control file (WSDUPD.INI) could not be created.

Example - Create a Program Manager group named COMPANY:

```
ADDGROUP "COMPANY"
```

NOTE: This function can be used with any third party shell program which emulates the Program Manager DDE interface.

ADDITEM [strvalue1] [strvalue2] [pathfile]

Parameter	Description and Notes
[strvalue1]	The group to which the item will be added.
[strvalue2]	The name of the new item.
[pathfile]	The .EXE file to be associated with the new item.

Description - Adds a new item to a Program Manager group.

Tip: When the ADDITEM script function is executed, the BrightWorks software distribution update program WSDUPD.EXE is automatically copied into the workstation's Windows directory. The WSDUPD.EXE command is also added to the "Load=" line in the WIN.INI file. The next time Windows is run at the workstation, the function is executed and WSDUPD.EXE is removed from the WIN.INI "Load=" line.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code if unsuccessful. The function might fail if WSDUPD.EXE could not be copied into the Windows directory or if the WSDUPD.EXE control file (WSDUPD.INI) could not be created.

Example - Create a Program Manager group named APPS, and then create a program icon within the new APPS group named EXCEL:

```
ADDGROUP "APPS"
ADDITEM "APPS" "EXCEL" "U:\MS\EXCEL\EXCEL.EXE"
```

NOTE: This function can be used with any third party shell program which emulates the Program Manager DDE interface. Also note that for this function the path specified will show up in the command line as well as the working directory. The EXCEL example above demonstrates this.

GETINIINT [pathfile] [strvalue1] [strvalue2] [intvar]

This function works in exactly the same way as GETINISTR (below) except it is used to retrieve integer values from INI files.

GETINISTR [pathfile] [strvalue1] [strvalue2] [strvar]

Parameter	Description and Notes
[pathfile]	The path and file name of the INI file.
[strvalue1]	The section of the INI file in which the entry is located (e.g., [386Enh]).
[strvalue2]	The entry whose associated string is to be retrieved (e.g., keyboard.drv=, however, do not include the = sign!).
[strvar]	Variable in which to place the found string. (Before being used as a parameter, this variable must be defined using the DEFINE function.)

Description - Retrieves a key value (string) from an INI file, and places the result in a variable.

Tip - If [strvalue2] is a null string or the key is not found, ADDDEVICE will add [strvalue1] in the position of the file indicated by [addopt].

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = -1 if the [strvalue2] section name does not exist

[RETVAL] = -2 if the [strvalue3] key does not exist

[RETVAL] = DOS error code in all other cases

Example - Determine whether Windows version 3.1 is installed at a workstation by looking at the CONTROL.INI file:

```
DEFINE "VER" STRING
GETINISTR "C:\WIN\CONTROL.INI" "[INSTALLED]" "3.1" VER
```

SCHEDULEWIN [path] [filename] [text]

Parameter	Description and Notes
[path]	The path to the file to be run.
[filename]	The file name to be run upon Windows execution.
[text]	Optional command line arguments for the file.

Description - Schedules a file to be run the next time the user runs Windows.

Tip - This function could be used to automate the installation of a Windows program if a macro playback utility is used. This function is also used for the inclusion of a Windows-based PowerScript.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code if unsuccessful. The function might fail if WSDUPD.EXE could not be copied into the Windows directory or if the WSDUPD.EXE control file (WSDUPD.INI) could not be created.

Example - Schedule the Notepad program to run the next time Windows is run, and also open the README.TXT notepad file:

```
SCHEDULEWIN "C:\WINDOWS" "NOTEPAD.EXE" "README.TXT"
```

WRITEINIINT [pathfile] [strvalue1] [strvalue2] [intvalue]

This function works exactly like WRITEINISTR (below), except that it is used to write an integer value to an INI file.

WRITEINISTR [pathfile] [strvalue1] [strvalue2] [strvalue3]

Parameter	Description and Notes
[pathfile]	The path and file name of the INI file.
[strvalue1]	The section in which [strvalue2] is located (e.g., [386Enh]).
[strvalue2]	The entry whose associated string is to be modified (e.g., keyboard.drv=, however, don't include the = sign!).
[strvalue3]	The string to be written to the INI file.

Description - Retrieves a key value (string) from an INI file, and writes the result to the INI file.

Tips:

- 1) If the section name specified in [strvalue1] is not found, then it will be added to the end of the INI file, with a new key=value added in that section.
- 2) If the [strvalue1] section is found but the key value specified in [strvalue2] is not found, the new key value is added directly after the section name [strvalue1].

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = DOS error code in all other cases

Example - Define a “medium priority” in the [SPOOLER] section of the WIN.INI file:

```
WRITEINISTR "C:\WIN\WIN.INI" "[SPOOLER]" "PRIORITY"
"MEDIUM"
```

Miscellaneous Functions

The Miscellaneous Functions include basic functions for defining, assigning, copying, comparing and concatenating variables.

NOTE: In the following function specifications, parameters in quotes represent literal parameters; all other parameters represent rules. Refer to "Rules and System Variables" on page 107 for a listing of the rules.

APPENDPATH [strvar] [strvalue]

Parameter	Description and Notes
[strvar]	The variable to contain the appended string (i.e., destination). (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[strvalue]	The string value to be appended (i.e., source).

Description - Adds a file name to a path or builds a path. This function acts the same way as STRCAT, except that it will check if the last character of [strvar] is a

“\”. If it is not, APPENDPATH will append a “\” to [strvar], and then [strvalue] will be appended. This is very useful (and necessary!) in building paths.

Return Value:

[RETV] = 0 always

Example - Define the variable named PATH to be a string-type. Copy the location of the network configuration files into the PATH variable and then append it to the C:\DRIVERS directory.

```
DEFINE "PATH" STRING
STRCOPY PATH [NETCFG]
APPENDPATH "C:\DRIVERS" PATH
```

ASSIGN [intvar] [intvalue]

Parameter	Description and Notes
[intvar]	The integer type variable name which will be assigned a value. (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[intvalue]	The numeric value to be assigned to the integer type variable.

Description - Performs a basic integer assignment operation (e.g., a = b).

Return Value:

[RETV] = 0 always

Example - Define the variable “NUM” as an integer type, and later assign 33 to the variable NUM:

```
DEFINE "NUM" INTEGER
ASSIGN NUM 33
```

DEFINE [text] [defineopt]

Parameter	Description and Notes
[text]	The variable being defined.
[defineopt]	The type of variable being defined (e.g., STRING or INTEGER).

Description - Used to create user defined variables of a string or integer type. This variable can then be used later in the script.

Tips:

- 1) All DEFINE statements must be declared before any script command is executed.
- 2) If a STRING type variable is declared, the login module will allocate 255 bytes (= 255 characters) of memory for the string. If an INTEGER type variable is declared, the login module will allocate 4 bytes (C type long which equals to an approximately -2 billion to +2 billion size integer).

Return Value:

[RETVAL] = 0 always

Example - Define the variable "ANSWER" as a string type.

```
DEFINE "ANSWER" STRING
```

EXIT [intvalue]

Parameter	Description and Notes
[intvalue]	An integer type variable.

Description - Ends the script.

Tips:

- 1) If [intvalue] is set to a non-zero value, then the login module will increment the error count by one for the upgrade package and note the log with the error number returned.
- 2) If the package has been defined to execute the script before decompressing the fileset, then the EXIT command will prevent the decompression of the fileset. (For more information on defining "Advanced Package Options," refer to the section "Creating and Editing Packages" in Chapter 9.)

Return Value: none

Example - End the script if an obtained value is greater than 50:

```
IF RESULT <= 50
  CFGSETVALUE "FILES" 55
ELSE
  EXIT 1
ENDIF
```

IF [intvalue1] [condoper] [intvalue2] ... {ELSE...} ENDIF

Parameter	Description and Notes
[intvalue1]	An integer type variable to be evaluated against [intvalue2].
[condoper]	Valid conditional operators are: =, !=, <, >, <=, >=
[intvalue2]	An integer type variable to evaluate [intvalue1] against.

Description - Allows conditional processing of functions. IF..THEN evaluates the conditional expression defined by [intvalue1] [condoper] [intvalue2]. If the condition evaluates to be TRUE, then all functions following THEN are executed until an ELSE or ENDIF is reached. If the condition evaluates to FALSE and ELSE is defined, then all functions following the ELSE are executed until an ENDIF is reached.

Tip - IFs can be nested up to 50 levels deep.

Return Value: none

Example - Obtain the FILES= value from the CONFIG.SYS file. If the value is less than or equal to 50, then change the value to 55; otherwise, exit the script:

```
DEFINE "RESULT" INTEGER
SETSYSFILE "C:\\" "CONFIG.SYS"
CFGGETVALUE "FILES" RESULT
IF RESULT <= 50
    CFGSETVALUE "FILES" 55
ELSE
    EXIT 1
ENDIF
```

NUMTOSTR [strvar] [intvalue]

Parameter	Description and Notes
[strvar]	The variable to contain the converted value. (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[intvalue]	The numeric value to be converted.

Description - Converts a numeric value to a string variable.

Return Value:

[RETVAL] = 0 always

Example - Convert the number 100 to a string and store the value in the defined variable named ONEHUNDRED:

```
DEFINE "ONEHUNDRED" STRING
NUMTOSTR ONEHUNDRED 100
```

PAUSE [text]

Parameter	Description and Notes
[text]	The text to be displayed on the user's screen during the pause. (This can be NULL)

Description - Pauses execution of the script until the user presses a key.

Tip - If [text] is NULL, then the default message "Strike any key to continue" is displayed on the screen.

Return Value = 0 always

Example - Display the message "Pausing... press any key to continue" during script execution.

```
PAUSE "PAUSING ... PRESS ANY KEY TO CONTINUE."
```

REBOOT

This function immediately reboots the user's PC. It does not accept any parameters and does not return any values. Before the reboot, the script file is closed, the log database is closed, and any necessary cleanup is performed.

NOTE: The PC will not reboot if a fileset is to be executed after the script. The REBOOT function might not work if the workstation is not 100% PC compatible.

SHELL [pathfile] {text} {shellopt}

Parameter	Description and Notes
[pathfile]	The path and file name to execute.
{text}	The file's optional command line arguments. (This can be NULL.)

{shellopt}	<i>Optional</i> argument which can only be either [KEEPPATH] or NULL.
------------	---

Description - Allows a user to execute an external DOS batch file, executable program, or DOS command. This function is used to include a DOS based PowerScript program.

Tip - To execute the program or batch file in [pathfile] and change to the specified path, use the KEEPPATH option as the {shellopt} parameter. If you don't specify the KEEPPATH option, SHELL will try to use the path from which the SDUPDATE program was run. KEEPPATH allows you to temporarily switch to the path from where you want to run the program.

Return Values:

[RETVAL] = 0 if successful

[RETVAL] = -1 if failed

Example - Execute LIST.COM and load the contents of the README.TXT file. Temporarily make the current directory C:\PUB\LIST.COM.

```
SHELL "C:\PUB\LIST.COM" "README.TXT" " " KEEPPATH
```

STRCAT [strvar] [strvalue]

Parameter	Description and Notes
[strvar]	The variable to contain the concatenated string (i.e., destination). (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[strvalue]	The string value to be appended (i.e., source).

Description - Appends the contents of [strvalue] to the end of the string [strvar].

Tip - If the resulting text in [strvar] is longer than the space allowed (255 bytes), then it will be truncated and properly null terminated.

Return Value:

[RETVAL] = 0 always

Example - Add the string "ADDTHIS" to a string variable named STRINGS1&2:

```
DEFINE "STRINGS1&2" STRING
STRCAT STRINGS1&2 "ADDTHIS"
```

STRCOMPARE [strvar] [strvalue]

Parameter	Description and Notes
[strvar]	The variable to be compared. (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[strvalue]	The value to compare the variable against.

Description - Does a byte for byte comparison of two strings.

Return Values:

[RETVAL] = 0 if the strings are identical

[RETVAL] = < 0 if [strvar] is less than [strvalue]

[RETVAL] = > 0 if [strvar] is greater than [strvalue]

Example - Check the current NetWare login name against a specified login name ("Supervisor").

```
DEFINE "NAME" STRING
STRCOPY NAME [LOGINNAME]
STRCOMPARE NAME "SUPERVISOR"
```

STRCOPY [strvar] [strvalue]

Parameter	Description and Notes
[strvar]	The variable to receive the copied string value (i.e., destination). (Before being used as a parameter, this variable must be defined using the DEFINE function.)
[strvalue]	The string value to be copied (i.e., source).

Description - Copies a value into a string, overwriting the previous contents of the string.

Return Value:

[RETVAL] = 0 always

Example - Copy the string "ABC" into the string variable named "HOLDABC":

```
DEFINE "HOLDABC" STRING
STRCOPY HOLDABC "ABC"
```

WRITELN [strvalue]

Parameter	Description and Notes
[strvalue]	The string to display on screen.

Description - Writes the [strvalue] line to stdout (e.g., the screen). This might be useful for displaying error messages, etc.

Return Value:

[RETVAL] = 0 always

Example - Define the variable named RESULT. Place the value of the FILES= statement in the CONFIG.SYS file into RESULT, and then write the value of RESULT.

```
DEFINE "VALUE" STRING
DEFINE "RESULT" INTEGER
SETSYSFILE "C:\\" "CONFIG.SYS"
CFGGETVALUE "FILES" RESULT
NUMTOSTR VALUE RESULT
WRITELN VALUE
```

Rules and System Variables

Rules

Most of the functions in the BrightWorks script language have parameters that are specified or passed to them. The valid entries for each parameter type are called *rules*. For example, the UPGRADEOS function has one parameter named [upgopt]. As indicated in the table below, the value of the [upgopt] parameter can be either 5.00 or 6.00. Therefore, the allowable values for the [upgopt] parameter are 5.00 and 6.00.

NOTE: When a user defined variable of string type is expected, [STRVAR] is the rule. When a user defined variable of integer type is expected, [INTVAR] is the rule.

The table below lists the rules (allowable values) for each parameter.

Rule Name	Allowed Values
ADDOPT	BEFORE AFTER
ATTRIBUTE	RO RW A SY H SH -A -SY -H -SH
CONDOPER	< > = != >= <=
DEFINEOPT	STRING INTEGER
DELETEOPT	ALL
FILENAME	[STRVAR] “filename.ext” (wild cards not allowed for file name)
FILEWILD	[STRVAR] “filename.ext” “*.*” (wild cards are allowed but not required for a file name)
INTVALUE	[INTVAR] [RETVAL] # (where # is a valid integer)
INTVAR	[INTVAR]
PATH	[STRVAR] “path” [TARGET] [BOOT_ROOT] [WINDIR] [WINSYSDIR] [NETCFG] [HDRIVE] [NDRIVE] [SERVERNAME] [LOGINNAME] [FUSIONNAME] [LOGSCRNAME]
PATHFILE	[STRVAR] “{path\}filename.ext”
SHELLOPT	KEEPPATH
STRVAR	[STRVAR]
STRVALUE	[STRVAR] “text” [TARGET] [BOOT_ROOT] [WINDIR] [WINSYSDIR] [NETCFG] [HDRIVE] [NDRIVE] [SERVERNAME] [LOGINNAME] [FUSIONNAME] [LOGSCRNAME]
TEXT	“text”
UPGOPT	5.00 6.00

System Variables

The rules listed in the above table are defined as follows:

String Type Rules:

Rule	Description
[BOOT_ROOT]	The root of the boot drive of the workstation on which the script is executed
[HDRIVE]	Drive letter of the first available hard drive (may be boot or network drive)
[FUSIONNAME]	Primary user name from BrightWorks databases (generally same as LOGINNAME)
[LOCATION]	Location field from BrightWorks inventory databases
[LOGINNAME]	Login name of user
[LOGSCRNAME]	Full path and file name of login script for user running update.
[NDRIVE]	Drive letter of the first available network drive
[NETCFG]	Path to NET.CFG used at NetWare shell load (must be in path)
[SERVERNAME]	Name of server on which the update program is running
[TARGET]	Installation path as defined by the administrator (or changed by user, if able to)
[WINDIR]	The user's Windows directory (directory in which the login module finds WIN.INI - this directory must be in the path)
[WINSYSDIR]	The user's Windows\System directory (directory in which the login module finds USER.EXE - this directory may be in the SYSTEM directory below WINDIR, or in the path)

Integer Type Rules:

Rule	Description
[DISKSPACE]	Available disk space in drive specified in [TARGETDIR]. The number is in bytes.
[RETVAL]	Return code of last command completed

DOS Error Codes

The following table lists the DOS error codes that may be returned from the script functions.

#	Message	Reason for Error	Action	Functions that Return the Error
2	File not found	A file specified in the script does not exist.	Check the filename and path.	GETINISTR() - The file from which you requested a string does not exist. GETINIINT() - The file from which you requested an integer does not exist.
3	Path not found	A directory path specified in the script does not exist.	Check the path and directory name.	DELETEDIR() - The directory that you requested to delete does not exist, or it does not exist in the location you specified.
4	Too many open files (no handles left)	Insufficient file handles specified in CONFIG.SYS	Increase the number of file handles in CONFIG.SYS.	COPY() All Easy System File and Windows System File functions.
5	Access denied	Unable to access the specified drive or file. Insufficient user rights, read only files, disk full.	Verify the user rights, file attributes and available disk space.	DELETEFILE(), ADDPATH(), ADDLINE(), REPLACELINE(), REPLACEKEY(), ADDDEVICE(), CFGSETVALUE(), CFGSETSTRING(), REPLACELINEADD(), WRITEINISTR(), WRITEINIINT() - Is the file flagged read only? Is the disk full? Does the end user have insufficient rights in the specified directory? DELETEDIR() - Subdirectories and/or files exist, and the "ALL" option was not used in the script. ADDGROUP(), ADDITEM(), SCHEDULEWIN() - Is WIN.INI flagged read only?
8	Insufficient memory	Not enough memory to complete the specified action.	Unload unnecessary TSRs, check workstation memory management.	All DOS, Easy System File and Windows System File functions.
15	Invalid drive	The drive specified does not exist.	Check the drives specified in the script.	All DOS, Easy System File and Windows System File functions.
16	Attempt to remove current directory	The directory you attempted to delete is active on a drive.		DELETEDIR() - Is the directory specified active on the drive? If it is a network drive, are other users active on the drive?

#	Message	Reason for Error	Action	Functions that Return the Error
17	Not same device	An action was specified on two separate drives.	Ensure that you are not attempting to “cross drives” on an action that does not permit this (e.g., RENAME)	RENAME() - Are the source and target locations different?
18	No more files	The specified file could not be found.	Check the path and filename. Check end user rights in the directory specified.	DELFIL(), ATTRIB(), RENAME(), SETSYSFILE(), COPY() - Does the specified file exist in the location specified? Does the end user have sufficient rights to see the file?
19	Disk is write protected	The write protect tab is enabled on the disk specified in the operation.	Remove the write protect tab.	All DOS, Easy System File and Windows System File functions.
21	Drive not ready	There is no disk in the drive specified in the operation.	Insert the diskette.	All DOS, Easy System File and Windows System File functions.
22	Invalid disk command	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
23	CRC error	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
24	Invalid length	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
25	Seek error	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
27	Sector not found	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
29	Write fault	Media access error.	Check the diskette or drive.	Bad or damaged diskette.

30	Read fault	Media access error.	Check the diskette or drive.	Bad or damaged diskette.
31	General failure	Media access error.	Check the diskette or drive.	Diskette may not be formatted.

Chapter 8 *Scopes*

Chapter 7 discussed the software distribution script language. This chapter discusses the creation and management of scopes—the group of workstations defined to receive a distributed package.

Introduction

A scope is a group of workstations defined to receive a distributed package. Defining a scope is as easy as assigning a name to the new scope. After the scope is created, any number of workstations can be included in the scope definition.

By taking advantage of the database of inventory information maintained by BrightWorks, users can create scopes by selecting from nodes that match specific filtering criteria. The filtering criteria is saved as a “query” and then applied against the database to narrow down the list of applicable workstations. Scope membership is subsequently assigned using the list of nodes that match the user-specified filtering criteria. Items such as CPU speed, workstation memory and installed software can be used to accommodate a scope’s intent. For example, a scope named CPU386 might consist of the network’s 386 workstations; a scope named 386>16MHz might consist of the network’s 386 workstations that also have a CPU clock frequency greater than 16 MHz.

Scopes and queries can be stored, used and reused as resources within BrightWorks. A user can create a new scope, as well as edit, copy, rename and delete a scope. The steps for managing both scopes and queries are provided in this chapter.

Creating Scopes

NOTE: At least one audit must have been run in order to create a scope.

Use the following procedure to create a new scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed listing the names of all defined scopes.

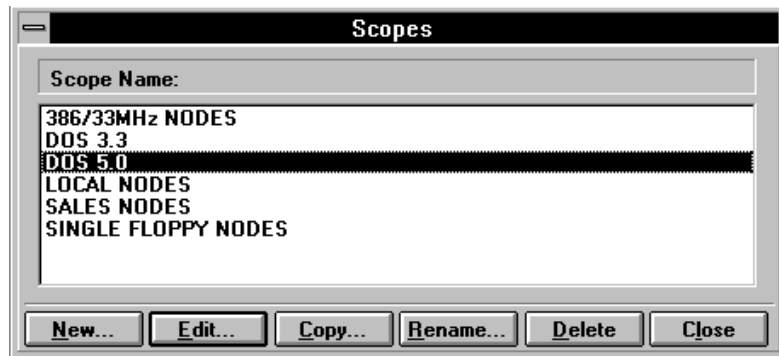


Figure 8-1: A List of Scopes

2. Choose New.

The New Scope dialog box is displayed prompting you to enter a name for the new scope.

3. Enter the new scope name.

A scope name can be up to 80 characters, and all typed characters are valid. For example, enter the new scope name “386/16MHZ” which will include the network’s 386/16MHz nodes.

4. Choose OK

The Edit Scope dialog box is displayed prompting you to define the new scope.

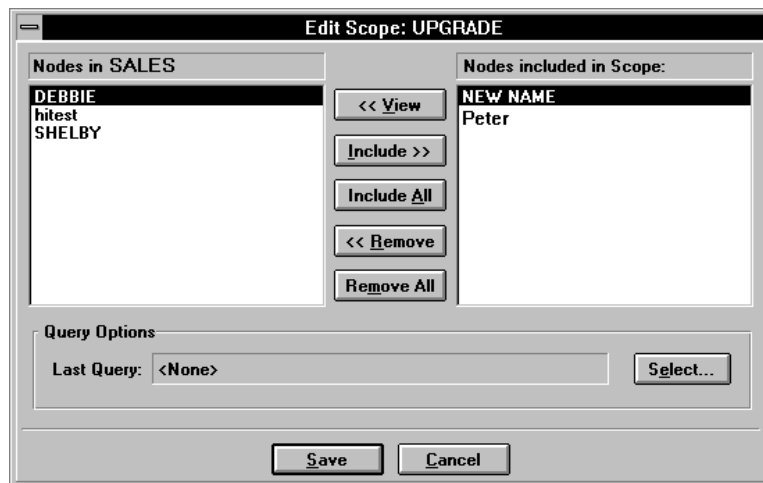


Figure 8-2: Editing Scope Members

The Edit Scope dialog box consists of two lists:

- The Nodes in SITE list on the left side of the dialog box consists of all node names that apply to the local site. The site name is indicated by the SITE text in the list title. (In Figure 8-2 above, SALES is the site name.) The nodes in this list are not included in the selected scope (i.e., the list to the right).
- The Nodes included in Scope list on the right side of the dialog box consist of the nodes that are in the selected scope.

NOTE: The Query Options section of the Edit Scope dialog box is used to perform a query to filter the node names in the Nodes in SITE list. For detailed instructions on performing queries, refer to the next section entitled “Scope Queries.”

5. Select the desired node from the Nodes in SITE list.
6. Choose Include.

The node moves to the Nodes Included in Scope list box. Choose Include All to move all moves.

If you want to remove a node, choose Remove. Choosing Remove All removes all nodes.

NOTE: Because user names can be edited through the Inventory Details dialog box, the node names in the Nodes in SITE list do not necessarily correspond to NetWare user names. As a result, there may be duplicate names in this list. Viewing the inventory details of nodes with the same name enables you to differentiate between the nodes. Choose View or double click on a name in either list to invoke the View Inventory Details dialog box for the selected node.

7. Choose Save.

Scope Queries

When no filtering criteria is applied to a scope, all nodes in the local BrightWorks site are listed in the Nodes in SITE list of the Edit Scope dialog box. This condition is indicated by the <None> entry in the Last Query field. Searching through a large list of nodes might make the process of defining scope membership cumbersome. Applying a query to a scope refines the number of nodes in the Nodes in SITE list so that scope membership can be made from a “qualified” list of nodes.

NOTE: Only one query can be applied to a scope at any time. Each query can consist of more than one filtering criteria. An applied query always filters from the entire list of nodes in the local BrightWorks site.

Queries can be saved and applied to any number of scopes. The same queries can be applied to BrightWorks inventory and distribution reports, as is discussed in Chapter 9 of *Using LAN Inventory*.

This section lists the procedures for:

- Applying a query to the scope
- Removing a query from the scope
- Creating a new query
- Editing a query
- Deleting a query

Applying a Query to the Scope

Use the following procedure to apply an existing query to a scope.

NOTE: This procedure assumes that you have already chosen Edit in the Scopes dialog box to display the Edit Scope dialog box.

1. Choose Select in the Query Options section of the Edit Scope dialog box.

The Select Query dialog box is displayed listing all defined queries.

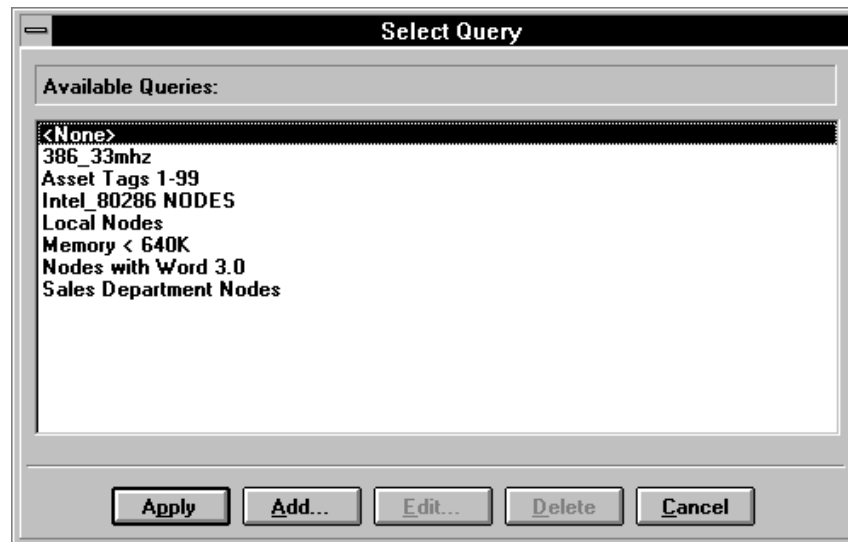


Figure 8-3: Selecting a Query to Apply to a Scope

2. Select the query name from the Available Queries list.

To select a query name, point to the query and click the left mouse button.

3. Choose Apply

NOTE: Applying a query to a scope causes a Printing Status dialog box to display while the database records are being sorted. When this occurs, nodes are being selected (i.e., the information is not being sent to the printer).

The Select Query dialog box closes, and the selected query name is placed into the Last Query field of the Edit Scope dialog box. The BrightWorks database records are sorted, and only the records that match the query's specified filter criteria will display in the Nodes in SITE list. Now you can assign scope members using a "qualified" list of nodes.

NOTE: If you already selected nodes to be included in the scope (i.e., the nodes listed in the Nodes Included in Scope list), the nodes continue to be "included" even if they do not match the filter criteria.

Removing a Query from the Scope

Use the following procedure to remove a scope query.

1. Choose Select in the Query Options group box of the Edit Scope dialog box.

The Select Query dialog box is displayed.

2. Select the <None> query name.

The Select Query dialog box closes. All nodes in the local BrightWorks site are listed in the Nodes in SITE list of the Edit Scope dialog box.

3. Choose Apply.

Creating a Complex Query

A query can consist of any number of filter entries that are defined to produce a desired result. The relationship between the filter entries is referred to as the "link."

Assume that you are responsible for upgrading the workstations of all network users in the Sales Department who are currently using Intel 286 machines with a CPU speed less than 16 MHz and which have 1.44 megabyte floppy disks designated as drive A.

Use the following procedure to create a query that defines the scope of users in the above example.

1. In the Add Query dialog box, enter a Query Name.

Enter a name that uniquely identifies this query (e.g., "Sales 286/16Mhz").

2. Define the first filter entry, and choose Insert.

Enter the following for each field:

- Component: Department
- Condition: =
- Description: Sales
- Query Link: AND

3. Define the second filter entry, and choose Insert.

Enter the following for each field:

- Component: Central Processing Unit
- Condition: =
- Description: Intel_80286
- Query Link: AND

4. Define the third filter entry, and choose insert.

Enter the following for each field:

- Component: CPU Clock Frequency
- Condition: <
- Description: 16MHz
- Query Link: AND

5. Define the fourth filter entry, and choose insert.

Enter the following for each field:

- Component: Floppy Disk #1
- Condition: =
- Description: A: 1.44 M
- Query Link: AND

6. Choose Save to save the query.

The filter entries in the Current Query list in the Edit Query dialog box should be identical to the following:

```
Department = Sales
Central Processing Unit = Intel_80286      AND
CPU Clock Frequency < 16MHz              AND
Floppy Disk #1 = A: 1.44 M                AND
```

Creating a New Query

NOTE: All queries are also available when generating BrightWorks inventory and distribution reports, as discussed in Chapter 9 of *Using LAN Inventory*.

Use the following procedure to create a new query.

NOTE: This procedure assumes that you have already chosen Select in the Edit Scope dialog box to display the Select Query dialog box.

1. Choose Add in the Select Query dialog box.

The Add Query dialog box is displayed. Press the TAB key to move from field to field within this dialog box.

Figure 8-4: Defining a New Query

2. In the Query Name text box, enter a query name.
A query name can be up to 80 characters in length.
3. Select a Component from the drop-down list box.
4. Select a conditional operator from the Condition drop-down list box.
Sample conditional operators include: equal to, less than, greater than, not equal to, etc.
5. Select a description of the component.

The items which automatically display in this list depend on the selected component. For example, “Intel_80386” might display if Central Processing Unit is entered in the Component field; “16.00 Mhz” might display if CPU Clock Frequency is entered in the Component field. Refer to the Note below.

6. Specify a filter entry.

The purpose of each filter entry is to narrow down the list of nodes that apply to the specified criteria. If more than one filter entry is defined, the entries are “linked” using either the AND or OR relationship.

For example, assume the following two filter entries:

Central Processing Unit = Intel_80386
CPU Clock Frequency > 66.00 Mhz

If the entries are linked with the OR relationship, only the nodes that satisfy *either* criteria (i.e., all Intel 80386 machines and all machines that have a clock speed greater than 66 Mhz) are included in the database sort.

If the entries are linked with the AND relationship, the nodes that satisfy *either* criteria (i.e., all 286 machines and all machines that have a 720k floppy drive) are included in the database sort.

Specify the relationship between the filter entries (e.g., Central Processing Unit = 80386 OR Central Processing Unit = 80486). The link options are AND and OR..

NOTE: The AND/OR links are logical links, and a formula must be carefully chosen to achieve the desired effect

7. Choose Insert to accept the filter entry definition.

The entry is added to the Current Query List in the Edit Query dialog box.

8. Repeat Steps 2 through 7 if required.

8. When all filter entries are defined, choose Save.

The query is saved and added to the Available Queries list in the Select Query dialog box. The new query can now be applied to a scope.

Editing a Query

Use the following procedure to edit the definition of an existing query.

NOTE: This procedure assumes that you have already chosen Select in the Edit Scope dialog box to display the Select Query dialog box.

1. Select a query from the Select Query dialog box, and choose Edit.

The Edit Query dialog box is displayed showing the defined filter entries for the query.

2. Modify the filter information.

To delete a filter entry, highlight the entry in the Current Query List and choose the Delete button.

To add a filter entry, specify the Component, Condition and Description, and choose the Insert button. (For detailed instructions on adding filter entries, refer to “Creating a New Query” above.)

3. Choose Save

NOTE: New filter entries are appended to the end of the Current Query List unless a filter entry is selected. If an existing filter entry is selected, then the new filter entry gets inserted above it when you choose Insert.

Deleting a Query

Use the following procedure to delete an existing query.

NOTE: This procedure assumes that you have already chosen Select in the Edit Scope dialog box to display the Select Query dialog box.

1. Select the desired query, and choose Delete.

A prompt displays asking you to confirm the deletion.

2. Choose Yes to delete the query.

If deleted, the query name is removed from the Available Queries list.

NOTE: Queries that are currently applied to a scope and/or BrightWorks inventory and distribution report can be deleted.

Managing Scopes

Editing Scopes

You may need to edit a scope for one of the following reasons:

- Existing scopes might need to be edited in order to add or delete members according to a change in a scope's intent.
- Scopes that are attached to packages might need to be edited when re-sending packages.

Scope *members* are the nodes that are defined as a group to receive a distributed package.

NOTE: An existing scope can be edited even if the scope is part of a scheduled package. This is useful if you need to re-send a package to a node(s). If new nodes are added to a scope that is included in an active package, then the package will be automatically distributed to the new nodes.

Use the following procedure to edit a scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed.

2. Select the desired scope and choose Edit.

A scope can also be selected for editing by double clicking on the scope name in the Scopes dialog box. The Edit Scope dialog box is displayed.

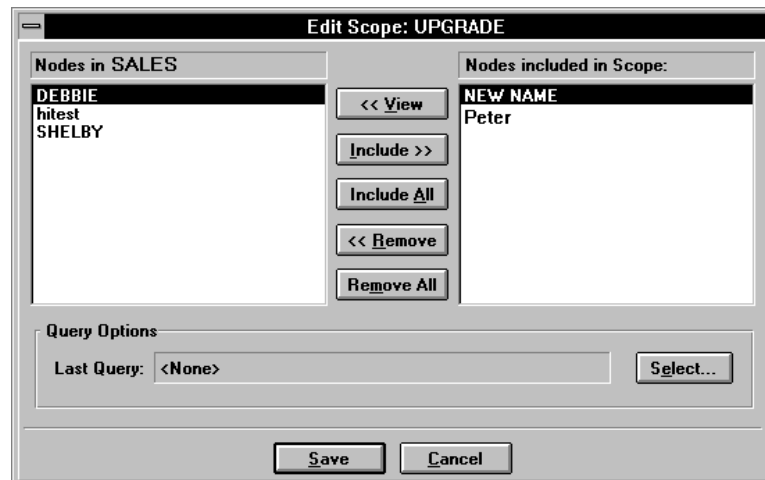


Figure 8-5: Editing Scope Membership

The Edit Scope dialog box consists of two lists:

The Nodes in SITE list on the left side of the dialog box consists of all node names that apply to the local site. The site name is indicated by the SITE text in the list title. (In Figure 8-5 above, SALES is the site name.) The nodes in this list are not included in the selected scope (i.e., the list to the right).

The Nodes included in Scope list on the right side of the dialog box consist of the nodes that are in the selected scope.

NOTE: The Query Options section of the Edit Scope dialog box is used to perform a query to filter the node names in the Nodes in SITE list. For detailed instructions on performing queries, refer to the section of this chapter entitled “Scope Queries.”

3. Select the desired node from the Nodes in SITE list.

4. Choose Include.

The node moves to the Nodes Included in Scope list box. Choose Include All to move all moves.

If you want to remove a node, choose Remove. Choosing Remove All removes all nodes.

NOTE: Because user names can be edited through the Inventory Details dialog box, the node names in the Nodes in SITE list do not necessarily correspond to NetWare user names. As a result, there may be duplicate names in this list. Viewing the inventory details of nodes with the same name enables you to differentiate between the nodes. Choose View or double click on a name in either list to invoke the Inventory Details dialog box for the selected node.

5. Choose Save.

Renaming Scopes

NOTE: A scope can be renamed even if it is part of an actively scheduled package.

Use the following procedure to rename a scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed.

2. Select the desired scope and choose Rename.

The Rename Scope dialog box is displayed prompting you to enter a new scope name.

3. Enter the new scope name.

The new scope name displays in the Scopes dialog box, and the old name is removed. All attributes of the old scope are preserved in the renamed scope (i.e., the scope members do not change).

4. Choose OK

Copying Scopes

NOTE: A scope can be copied even if the original scope is part of an actively scheduled package.

Use the following procedure to copy a scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed.

2. Select the desired scope and choose Copy.

The Copy Scope dialog box is displayed prompting you to enter a name for the new scope.

3. Enter the new scope name.

The new scope name is added to the Scopes dialog box. The new scope members are identical to the original scope members.

4. Choose OK.

Deleting Scopes

NOTE: A scope that is part of a scheduled package cannot be deleted.

Use the following procedure to delete a scope.

1. Choose Tools | Scopes.

The Scopes dialog box is displayed.

2. Select the desired scope and choose Delete.

A prompt displays asking you to confirm the deletion.

3. Choose Yes to delete the scope.

If deleted, the scope name is removed from the Scopes dialog box.

Chapter 9 Packages

Chapter 8 discussed the creation and management of scopes. This chapter discusses the creation and management of packages—the method by which software is distributed across your network.

Introduction

Creating and activating a package is the method by which software is distributed across your local area network. When a package is created, it is assigned a scope and a “Start Date.” Upon reaching the start date and running the SDUPDATE.EXE program at a workstation in the scope, an active package is automatically sent to the workstation.

A package also consists of a fileset and/or a script to be distributed to the group of remote workstations. For example, a package named UPGRADE_DOS might include a scope of users running DOS 3.3. The package might also include a script which installs a fileset consisting of the DOS 6.0 files.

The software distribution update program (SDUPDATE.EXE) is integrated with the packaging functionality. The update program is responsible for determining the conditions for accepting or rejecting a package. The program is also responsible for reporting on package status as input to the Packages window and the Software Distribution Log.

Package Features

In addition to the contents and scope, every package is assigned a schedule and a method for delivery. A package’s schedule is the date on which the package is to be distributed. The method of delivery specifies instructions for the receiving workstation. There are several options available to the workstation user when a package is received. For example, package ABC might be configured to prompt the user five times to accept the package before proceeding with the installation of its fileset.

An administrator can create a new package, as well as edit, rename and delete a package. The steps for each procedure are discussed in this chapter.

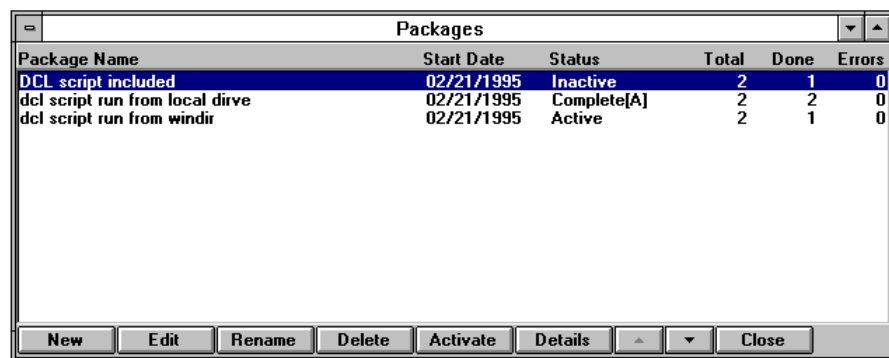
Creating and Editing Packages

Creating Packages

Use the following procedure to create a new package.

1. Choose Tools | Packages or choose the Distribute tool bar button.

The Packages window is displayed.



Package Name	Start Date	Status	Total	Done	Errors
DCL script included	02/21/1995	Inactive	2	1	0
dcl script run from local dirve	02/21/1995	Complete[A]	2	2	0
dcl script run from windir	02/21/1995	Active	2	1	0

Buttons: New, Edit, Rename, Delete, Activate, Details, Close

Figure 9-1: Packages Window

This window lists the names of all defined packages. For each package, the following is indicated:

Field	Description
Start Date	The date the package will be distributed
Status	The package's status (Active or Inactive)
Total	The total number of workstation in the package's scope
Done	The number of successful updates so far
Errors	The total number of failed attempts at performing an update

NOTE: Packages that have the same Start Date are distributed in the order in which they appear in the Packages window. To change a package's priority, refer to "Prioritizing Packages" on page 132.

2. Choose New in the Packages window.

The New Package dialog box is displayed prompting you to enter a name for the new package.

3. Enter the new package name.

The package name can be up to 80 characters, and all typed characters are valid.

4. Choose OK.

A New Package dialog box is displayed. The name of the new package is indicated in the title bar of the dialog box.

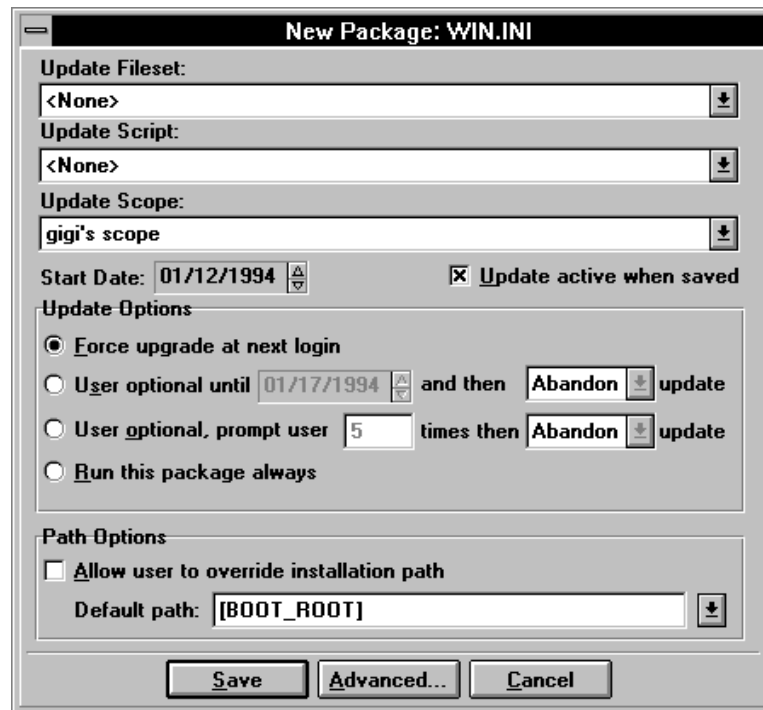


Figure 9-2: Creating a New Package

5. Select a fileset and/or a script from the provided list boxes to be included in the package.

NOTE: A package must include either one fileset or one script, or both.

6. Select the scope to receive the package from the list box.

NOTE: A scope must be assigned to the package.

7. Use the spin control to assign the package's Start Date.

Enter the date on which the package is to be distributed. The current date appears as a default in this field. Use the up/down arrow buttons to the right of the Start Date field to scroll to the desired date, or press the F4 key to display a calendar from which a date can be selected.

8. Select the "Update active when saved" check box if desired.

This option automatically places the package in an active state upon saving the package. (An active package will get distributed automatically on its assigned start date.)

If this field is not selected, an Inactive status will be assigned to the package. An inactive package will not get distributed automatically on its assigned start date.

8. Select the appropriate radio button in the Update Options group box.

The selected Update Option instructs the software distribution update program how it should interact with the receiving workstation user at login time.

The four Update Options are:

Option	Description
Force upgrade at next login	This option forces the package's receipt on the user at the next login. If an error occurs, the distribution is halted so the administrator can address the problem and reschedule the package.
User optional until [DATE] and then [ABANDON, FORCE] update	This option allows the user to refuse the package until the indicated DATE. After the DATE, the package is either abandoned or forced to be received by the user.
User optional, prompt user [# TIMES] and then [ABANDON, FORCE] update	This option allows the user to refuse the package a certain number of times (# TIMES). After the threshold is reached, the package is either abandoned or forced to be received by the user.
Run this package always	This option forces the package's receipt on the user at each and every login. This update option is most useful in situations where system variables are being modified directly by users.

10. Specify the target path in which the fileset should be decompressed.

A default path must be assigned to the package. The default path is the target path in which the distributed software (i.e., fileset) is to be installed or copied. The default path can be:

- entered as a hard-coded drive mapping and directory combination (e.g., C:\BIN\DRIVERS).

- entered as a hard-coded server, volume and directory combination (e.g., SERVER/VOLUME:\DIR or VOLUME:\DIR). If a server is specified, then the user receiving the package must be attached to the server.
- reliant upon a system configuration found at the receiving workstation. Reliance is implemented by using one of the following target default path options available from the drop down list associated with this field:

Option	Description
[BOOT_ROOT]	The root of the receiving machine's boot disk
[HDRIVE]	The receiving machine's first hard drive
[NDRIVE]	The receiving machine's first network drive
[NETCFG]	Path to the receiving machine's network configuration (which must be in the path)
[WINDIR]	The receiving machine's Windows directory (the directory in which the login module finds WIN.INI - this directory must be in the path)
[WINSYSDIR]	The receiving machine's Windows\System directory (directory in which the login module finds USER.EXE - this directory may be in the SYSTEM directory below WINDIR, or in the path)

These variables can be used in combination with a hard-coded value (e.g., [WINSYSDIR]\TEMP). In this case, the backslash character (\) is required and the variable name must be first. If a specified directory does not exist, it will be created.

If desired, check the "Allow user to override installation path" option to allow the user at the receiving workstation to override the selected path.

11. To define advanced package options, choose Advanced. (Otherwise skip to step 16.)

The Advanced Options dialog box is displayed.

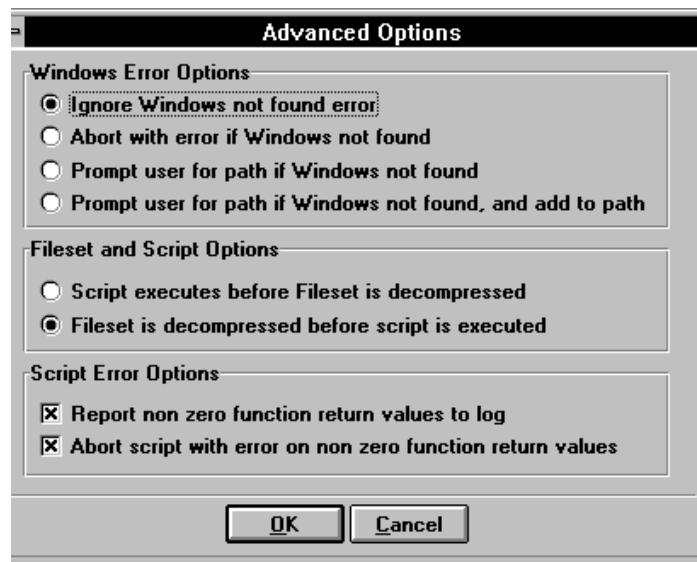


Figure 9-3: Selecting Advanced Options

12. Select the desired radio buttons in the Windows Error Options group box.
Choose from:
 - Ignore Windows not found error
 - Abort with error if Windows not found
 - Prompt user for path if Windows not found
 - Prompt user for path if Windows not found, and add to path
13. Select the desired radio buttons in the Fileset and Script Options group box.
Choose from:
 - Script executes before Fileset is decompressed
 - Fileset is decompressed before script is executed
14. Select the desired radio buttons in the Script Error Options group box.
Choose from:
 - Report non zero function return values to log
 - Abort script with error on non zero function return values
15. Choose OK to save settings and return to the New Packages dialog box.

NOTE: Several script functions may return a non-zero value that is not considered to be an error. For example, the FINDFILE function returns a -1 if the file is not found; the STRCOMPARE function returns non-zero value if the two strings are not equal. You might want to disable the Script Error Options when using functions that return non-zero values even when successful.

16. When all package attributes are defined, choose Save.

A completed New Package dialog box may look similar to Figure 9-4.

Figure 9-4: Editing Packages

Editing a package may be necessary in order to modify package attributes.

Use the following procedure to edit the attributes of a package.

1. Choose Tools | Packages.

The Packages dialog box is displayed.

2. Select the desired package and choose Edit.

A package can also be selected for edit by double clicking on the package name in the Packages window. The Edit Package dialog box is displayed showing the configuration of the selected package. The fields and options in this dialog box are identical to the New Package dialog box (Figure 9-4 above).

The name of the package being edited displays in the title bar of the Edit Package dialog box.

3. Modify the package attributes.

Changes can be made to any field except the Update Scope, Update Fileset and Update Script fields.

NOTES: a - Although the package's scope, fileset and script cannot be changed, their definition can be changed. For example, if Scope ABC is assigned to the package, you cannot assign Scope XYZ to the package but you can edit the members of scope ABC.

b - If a package fails, it can be re-distributed to a user by first removing the user from the scope and saving the edited scope, and then adding the user back into the scope and again saving the edited scope. Editing a scope changes all instances of the scope (i.e., even those included in packages).

If the distribution of a package has already begun, the changes made to the package take effect on the next node in the scope to receive the package.

4. Choose Save.

The Packages window re-displays.

NOTE: To display the Software Distribution Log History details associated with a package, highlight the package in the Packages window and choose Details. The Log Details dialog box is displayed showing the details for the selected package. This dialog box is discussed in "The Software Distribution Log" in Chapter 10.

Managing Packages

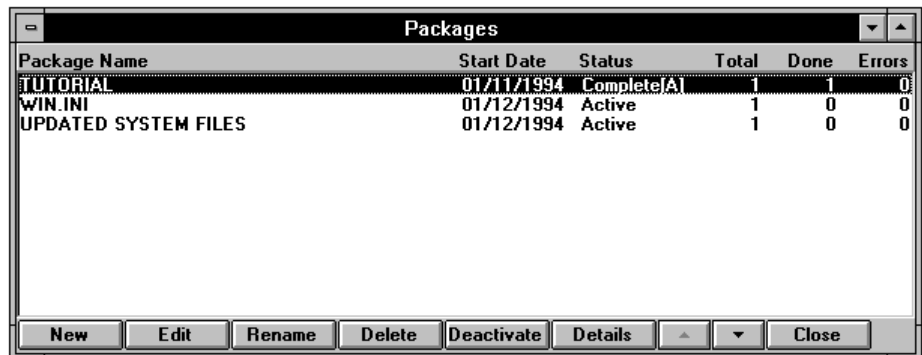
Prioritizing Packages

The priority with which each package is run is determined by its position in the Packages window. Packages that have the same Start Date are distributed in the order in which they are listed.

To modify a package's run time, highlight the package in the Packages window, and choose either the Up or Down button. The highlighted package will be moved in the selected direction.

For example, in Figure 9-5, there are two packages scheduled to be distributed on 1/12/1994. Because of their position in the list of packages, WIN_INI will be distributed before UPDATED SYSTEM FILES. To change their distribution order, highlight the UPDATED SYSTEM FILES fileset and choose the up arrow button.

Also note that the status of the TUTORIAL package is “Complete (A).” The (A) indicates that although the package has been distributed to all workstations in the scope, the package is still active. Therefore, if the package’s scope is edited to include additional nodes, the package will automatically get distributed to those new nodes.



Package Name	Start Date	Status	Total	Done	Errors
TUTORIAL	01/11/1994	Complete(A)	1	1	0
WIN.INI	01/12/1994	Active	1	0	0
UPDATED SYSTEM FILES	01/12/1994	Active	1	0	0

Buttons: New, Edit, Rename, Delete, Deactivate, Details, Close

Figure 9-5: Renaming Packages

Changing the name of an existing package renames the package in the Packages window.

NOTE: Actively scheduled packages can be renamed.

Use the following procedure to rename a package.

1. Choose Tools | Packages.

The Packages dialog box is displayed.

2. Select the desired package and choose Rename.

The Rename Package dialog box is displayed prompting you to enter a new package name.

3. Enter the new package name.

The new package name displays in the Packages window, and the old name is removed. All attributes of the old package are preserved in the renamed package (i.e., the package’s scope, script and/or fileset definitions do not change).

4. Choose OK.

Activating/Deactivating Packages

New packages are assigned the Active status if the Update Active when Saved option is selected. An active package is automatically distributed upon reaching its assigned Start Date. An inactive package will not be distributed until it is re-activated.

The status indication of a selected package in the Packages window toggles between Active/Inactive by choosing the Activate/Deactivate toggle button in the Packages window.

The status of a completed package (i.e., a package that has been sent to all nodes in its scope) remains active. As in Figure 9-5 above, its status is indicated by “Complete (A).” By highlighting the completed package and choosing the Deactivate toggle button, the status changes to “Complete (I)” which indicates an inactive status.

Deleting Packages

Use the following procedure to delete a package.

NOTE: Actively scheduled packages cannot be deleted. To delete a package with an Active status, first make the package status Inactive by highlighting the package and choosing Deactivate. Then, perform the Delete action.

1. Choose Tools | Packages.

The Packages window is displayed.

2. Select the desired package, and choose Delete.

A prompt displays asking you to confirm the deletion.

3. Choose Yes to delete the package.

If deleted, the package name is removed from the Packages window.

NOTE: Deleting a package does not delete the associated log entry in the Software Distribution Log History dialog box.

The Package Timer

The Packages window is displayed the status of each scheduled package. Status information includes the number of workstations in the package’s scope (Total), the current number of successful updates (Done) and the total number of failed attempts

at performing an update (Errors). The window contents are updated according to a defined package timer interval.

Setting the Package Timer Interval

Use the following procedure to set the package timer and define the interval at which the Packages window data is updated.

1. Choose Configure | Set Package Timer.

The Set Package Timer dialog box is displayed, as in Figure 9-6.



Figure 9-6: Setting the Package Update Interval

2. Enter the time interval at which the Packages window should be updated.

Enter the time in seconds. (The default is 30 seconds.) The information in the Packages window will be updated at the defined interval.

3. Choose OK.

NOTE: The status of the Packages window can also be updated on demand by choosing Configure | Set Refresh Timer.

NOTES

Chapter 10 Monitoring Software Distribution

Chapter 9 discussed the creation and management of packages. This chapter discusses the procedures for using the Software Distribution Log History dialog box to monitor the success of distributed packages.

Introduction

The chronological events associated with scheduled packages can be viewed from the Software Distribution Log History dialog box. The log history provides the information necessary for monitoring the success or failure of a distributed package. For example, if package WINUPDATE is distributed to a scope having three members, the installation of the software and update details for all three receiving workstations can be verified through the log history details.

The Software Distribution Log

Viewing Log History Details

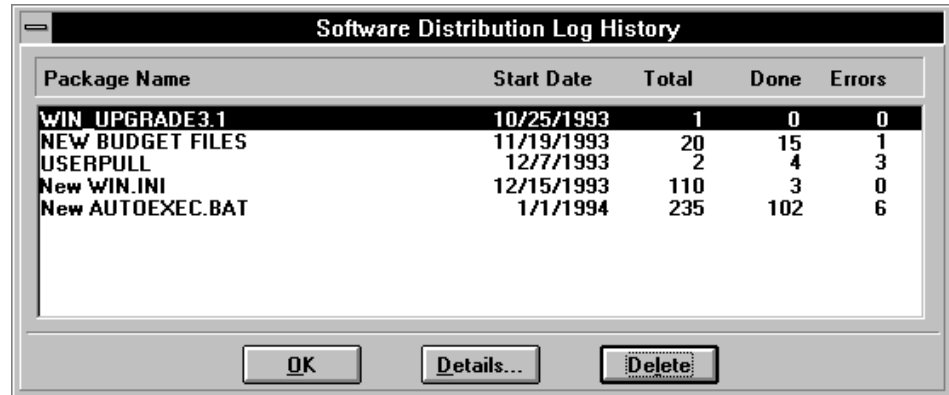
Use the following procedure to monitor and maintain the log history of distributed packages.

1. Choose View | Distribution Log.

The Software Distribution Log History dialog box is displayed.

The history log provides a summary of all scheduled packages. The following information is provided for each package:

Option	Description
Start Date	The date the package will be distributed
Total	The total number of workstation in the package's scope
Done	The number of successful updates so far
Errors	The total number of failed attempts at performing an update



The dialog box titled "Software Distribution Log History" contains a table with the following data:

Package Name	Start Date	Total	Done	Errors
WIN UPGRADE3.1	10/25/1993	1	0	0
NEW BUDGET FILES	11/19/1993	20	15	1
USERPULL	12/7/1993	2	4	3
New WIN.INI	12/15/1993	110	3	0
New AUTOEXEC.BAT	1/1/1994	235	102	6

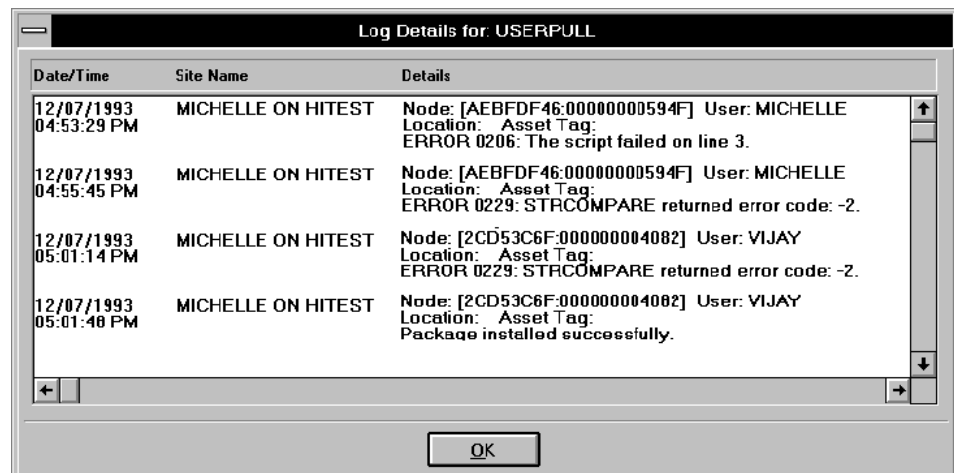
At the bottom of the dialog box are three buttons: "OK", "Details...", and "Delete".

Figure 10-1: Software Distribution Log

NOTE: The completed updates (Done column) may be greater than the number of users scheduled to receive a package (Total column). This can occur as a result of rescheduling packages or when using the “Run this package always” update option during package scheduling.

- To view the individual events of a package, select the package from the Software Distribution Log History dialog box, and choose Details.

A package can also be selected by double clicking on the package name in the Software Distribution Log History dialog box. The Log Details dialog box for the selected package displays.



The dialog box titled "Log Details for USERPULL" displays a list of events with the following data:

Date/Time	Site Name	Details
12/07/1993 04:53:29 PM	MICHELLE ON HITEST	Node: [AEBFDF46:00000000594F] User: MICHELLE Location: Asset Tag: ERROR 0206: The script failed on line 3.
12/07/1993 04:55:45 PM	MICHELLE ON HITEST	Node: [AEBFDF46:00000000594F] User: MICHELLE Location: Asset Tag: ERROR 0229: STRCMPARE returned error code: -2.
12/07/1993 05:01:14 PM	MICHELLE ON HITEST	Node: [2CD53C6F:000000004082] User: VIJAY Location: Asset Tag: ERROR 0229: STRCMPARE returned error code: -2.
12/07/1993 05:01:48 PM	MICHELLE ON HITEST	Node: [2CD53C6F:000000004082] User: VIJAY Location: Asset Tag: Package installed successfully.

At the bottom of the dialog box is an "OK" button.

Figure 10-2: Package Distribution Details

In addition to the distribution Date/Time and target Site Name, the Details of the package's chronological events are shown in three lines:

- Identification of the target workstation - node address, user name
- Additional target workstation information - location, asset tag
- Results - some possible results are:
 - Package installed successfully.
 - Error [#]: Script "[Script name]" has not been completed successfully.

NOTE: The error numbers related to unsuccessful script execution are documented in Chapter 7, "Software Distribution Script Language." Non-zero error numbers only display if the corresponding option was selected when the package was created. For instructions on defining advanced package options, refer to "Creating and Editing Packages" in Chapter 9.

3. To delete a log entry, select the package from the Software Distribution Log History dialog box, and choose Delete.

The package is removed from the list of scheduled packages.

NOTE: A log entry cannot be deleted if its associated package still exists.

4. Choose OK to close the Software Distribution Log History dialog box.

NOTES

Appendix A Error Messages

This appendix lists all error messages associated with BrightWorks. The error messages are listed alphabetically.

For error messages for the metering and inventory capabilities, refer to Appendix A of *Using SiteMeter* and Appendix B in *Using LAN Inventory*, respectively.

Script Editing Error Messages

The following error messages may be displayed while using BrightWorks' script editing features.

Another script is already using this file.

You must enter a unique file name for each script.

Do you want to save the changes?

You have tried to close the script editor window without saving your changes. Choose Yes to save your editing changes and close the script editor, or choose the No button to cancel your changes.

Duplicate names are not allowed.

A script with this name already exists. Enter a unique name.

File does not exist.

The script source file cannot be found. Verify the drive and directory on which you are searching.

File exceeds capacity of this editor.

A script source file cannot exceed 64k in size.

Source and target file names are the same.

You must enter a target name that is different than the script source file name.

Text Not Found.

The string being searched for via the Find String dialog box cannot be found.

There is no entry selected.

You must first select a script before choosing the delete or rename options.

This script cannot be deleted because it is a part of a scheduled package.

You cannot delete a script that is part of an actively scheduled package.

Unable to create new file.

The new script source file cannot be created. Verify that you have sufficient rights to create the new file.

You cannot edit more than eight documents at a time.

You can have a maximum of eight script editor windows open.

Software Distribution Error Messages

The following error messages may be displayed while using BrightWorks' distribution features.

node(s) have been deleted from the Inventory since the last time this scope was edited.

These nodes will be deleted from this scope and the corresponding packages upon saving.

Some nodes have been deleted from your baseline since the last time you modified this scope. These nodes will be deleted from your scope and all packages this scope is attached to when you save the scope.

A site must be defined before this feature may be utilized.

You must define a Site before using any of the distribution features (Packages, Scopes, etc..).

An error occurred compressing fileset.

An internal compression error occurred. The file(s) in the fileset may be corrupt.

An error occurred copying a file.

An error occurred while copying a fileset. Check your rights and the available disk space.

An invalid path to filesets was detected. It has been reset to the current directory.

This is an informational message. It generally occurs if you change the drive mapping from when BrightWorks was originally installed. Use the Administration/Distribution/Preferences menu commands to set the fileset path to the desired location.

Could not create Fileset "<name>".

The fileset could not be created. Check your rights and available disk space.

Could not create the target file.

Unable to create the new fileset during a copy. Check your rights and the available disk space.

Could not locate the Fileset for this package.

Either the fileset has been deleted, or the path to the filesets has been changed. Verify the path to filesets from Administration/Distribution/Preferences menu commands.

Error allocating memory.

General error indicating that not enough memory and/or system resources exist to accomplish a particular task. Close one or more applications and retry the operation.

Error renaming Fileset.

The fileset could not be renamed. Check your rights.

Error updating: <name>.

Unable to write to the fileset. Check your rights.

Insufficient memory available to compress files. Close one or more applications and try again.

Close one or more applications to free system resources/memory.

Name cannot be left empty.

You must enter a valid name for a fileset, script, or scope.

The fileset path is invalid. Only a valid network drive may be chosen.

You cannot store filesets on your local hard drive because your users who run SDUPDATE will not be able to locate them. Store all filesets in a location from which all users have access.

There are no log entries to display.

No users have run SDUPDATE for this package yet.

This Fileset cannot be deleted because it is part of a scheduled package.**This Scope cannot be deleted because it is part of a scheduled package.****This Log entry cannot be deleted because it is part of a scheduled package.****This Script cannot be deleted because it is part of a scheduled package.**

You must first delete any packages a scope, script fileset or log entry is part of before deleting that item.

This scope has been deleted.

This scope was deleted (probably by another user) before you attempted to edit it.

Unable to delete "<name>".

The fileset could not be deleted. Check your rights.

Unable to determine Fileset name. You may be running low on memory.

Close one or more applications to free up system resources/memory.

Unable to locate inventory equipment record. This node has probably been deleted.

Another user probably deleted the specified user from your baseline.

Unable to open source file.

Could not open the source fileset for copy. Check that it exists and that you have the appropriate rights.

Unable to read files in this Fileset. The Fileset is probably corrupted.

An unexpected 'End Of File' was encountered in the fileset. Delete the fileset and create it again. This fileset was corrupted.

Write error while compressing fileset. Check to be sure that you have sufficient rights and enough disk space.

The fileset could not be compressed. Check your rights and available disk space.

You cannot delete a package that is active.

Deactivate the package, and then delete it.

You must create a Scope before you can save a package.

Because a package must consist of one scope, at least one scope must be created before creating a package.

You must select a Fileset and/or a script for this package.

Select either a fileset and/or a script to run for this package.

You must select a Scope that has at least one node.

A scope of at least one node is required in order to save a package.

You must specify a default installation path.

You must specify a path in which to install the fileset and to which the [TARGET] system variable should default.

Software Distribution Update Program Error Messages

The following error messages may display while running BrightWorks' distribution Update program (SDUPDATE.EXE). The messages are listed in error number order.

ERROR 0101: Error determining boot drive letter.

This is an internal DOS error. You may be using an old DOS version. DOS 3.x is required.

ERROR 0103: Error determining first available network drive.

Check the connection to the file server, and retry the operation.

ERROR 0104: Error determining first available hard drive.

This is an internal DOS error. You may be using an old DOS version. DOS 3.x is required.

ERROR 0105: Not enough memory to create system variable: <VariableName>.

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0106: There is no Inventory Site in the specified drive.

Before a package can be installed, you must run EQUIP on the workstation which is to receive the distributed package. You must also run an audit to include that workstation in the baseline. Be sure that you are specifying the proper path to the BrightWorks inventory database file.

ERROR 0107: Floppy Disk Error: XX

Check that the disk is in the drive, is write enabled, and is formatted properly.

ERROR 0108: Drive 'X:' not ready or invalid drive.

Check that the disk is in the drive, is write enabled, and is formatted properly.

ERROR 0109: EQUIP needs to be run first, or your disk is write protected.

Before a package can be installed, you must run EQUIP on the workstation which is to receive the distributed package. You must also run an audit to include that workstation in the baseline. Be sure that you are specifying the proper path to the BrightWorks inventory database file.

ERROR 0110: This PC isn't in the BrightWorks Inventory database!. Be sure to run an audit prior to running SDUPDATE!

Before a package can be installed, you must run EQUIP on the workstation which is to receive the distributed package. You must also run an audit to include that workstation in the baseline. Be sure that you are specifying the proper path to the BrightWorks inventory database file.

ERROR 0111: There is no drive letter specified in default path: <Path>

The default path that was specified for this package is invalid. It must be in the form of "d:\[path]", "SERVER/VOLUME:\[path]", "VOLUME:\[path]", or one of the pre-defined system variables present in the combobox. D:\ is the drive letter, and [path] is the optional path. The user running SDUPDATE must be attached to SERVER, VOLUME must be a valid volume on that server, and the user must have rights to that volume.

ERROR 0112: An invalid drive letter was specified in default path: <Path>

The default path that was specified for this package is invalid. It must be in the form of "d:\[path]", "SERVER/VOLUME:\[path]", "VOLUME:\[path]", or one of the pre-defined system variables present in the combobox. D:\ is the drive letter, and [path] is the optional path. The user running SDUPDATE must be attached to SERVER, VOLUME must be a valid volume on that server, and the user must have rights to that volume.

ERROR 0113: Determining drive mapping to SERVER/USER in default path: <Path>

The default path that was specified for this package is invalid. It must be in the form of "d:\[path]", "SERVER/VOLUME:\[path]", "VOLUME:\[path]", or one of the pre-defined system variables present in the combobox. D:\ is the drive letter, and [path] is the optional path. The user running SDUPDATE must be attached to SERVER, VOLUME must be a valid volume on that server, and the user must have rights to that volume.

ERROR 0114: Error creating default path: <Path>

The default path that was specified for this package is invalid. It must be in the form of "d:\[path]", "SERVER/VOLUME:\[path]", "VOLUME:\[path]", or one of the pre-defined system variables present in the combobox. D:\ is the drive letter, and [path] is the optional path. The user running SDUPDATE must be attached to SERVER, VOLUME must be a valid volume on that server, and the user must have rights to that volume.

ERROR 0115: Not attached to file server: <ServerName>

The user running SDUPDATE must be attached to the server.

ERROR 0116: NetWare error: 0xXXXX

NetWare error. Check connection to server, and verify your rights.

ERROR 0117: Invalid SERVER/VOLUME format in default path: <pathname>

The default path that was specified for this package is invalid. It must be in the form of "d:\[path]", "SERVER/VOLUME:\[path]", "VOLUME:\[path]", or one of the pre-defined system variables present in the combobox. D:\ is the drive letter, and [path] is the optional path. The user running SDUPDATE must be attached to SERVER, VOLUME must be a valid volume on that server, and the user must have rights to that volume.

ERROR 0118: No such volume: <VolumeName>

The update program is unable to locate the specified volume. Be sure the user running SDUPDATE has rights to the volume and that the volume exists.

ERROR 0119: There are no available drive letters to map a drive to.

The user running SDUPDATE's drive map table is full. Delete one or more drive mappings for this user.

ERROR 0120: Windows must be in your path to install this package. Aborting package install.

This package requires that SDUPDATE can locate Windows in order to install. Add the Windows directory to your path.

ERROR 0121: Unable to open Fileset: <FilesetName>

Could not open the fileset. It was either deleted or the user has no rights to the fileset path.

ERROR 0122: Unable to allocate buffers for Fileset: <FilesetName>

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0123: File <FilesetName> is not a valid fileset!

SDUPDATE found the file, but it is not a valid fileset. From the BrightWorks console, choose the Fileset command from the Tools menu to create and edit filesets.

ERROR 0124: Unable to create file: <FileName>

Could not create a file. Check your rights and the available disk space.

ERROR 0125: Error in Fileset: <FilesetName>

An unexpected End of File was encountered in the fileset. Delete the fileset and create it again. This fileset was corrupted.

ERROR 0126: Script "<ScriptName>" has not been compiled!

The script for this package was modified and attempted to have been compiled, but the compile failed.

ERROR 0127: File <filename> doesn't exist or isn't in path.

Could not locate the specified file.

ERROR 0128: Out of memory.

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0129: Out of disk space decompressing: <FilesetName> to <DefaultPath>.

Out of space while decompressing the fileset. This test is done before any files are decompressed.

ERROR 0201: Unable to initialize Btrieve handler.

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0204: Unable to locate script data file: <ScriptFile>!

The compiled script file could not be located. It was probably deleted. Re-compile the script.

ERROR 0206: The script failed on line XX.

This message displays any time the script fails while executing. The message references the line number on which the script failed.

ERROR 0211: <FunctionName> didn't have enough memory to create a variable.

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0214: Maximum nest count reached processing function: IF.

IF...THEN functions can only be nested 50 levels deep.

ERROR 0216: Invalid drive letter specified in function: <FunctionName>

A valid path is: "d:\path" where *d* is a valid drive letter and *path* is a valid path.

ERROR 0217: Invalid path specified in function: <FunctionName>.

A valid path is: "d:\path" where *d* is a valid drive letter and *path* is a valid path.

ERROR 0218: Function UPGRADEOS requires DOS boot files on your boot disk!

The disk in your boot drive does not have any DOS system files. These files are necessary for the UPGRADEOS function to proceed.

ERROR 0219: Function UPGRADEOS needs the DOS files to perform the upgrade.

Be sure to run EQUIP on a DOS 5.0 workstation, a DOS 6.0 workstation, and a DOS 6.2 workstation prior to using the UPGRADEOS function for that DOS version. EQUIP picks up DOS system information and saves it in the path where the inventory databases are located, so SDUPDATE can find them and use them to upgrade. You should also be sure you have the appropriate license for the DOS version you are installing.

ERROR 0220: Out of memory in function: <FunctionName>

Ran out of memory. Unload some TSRs and/or device drivers.

ERROR 0221: UPGRADEOS was unable to delete system files from your boot disk.

The boot disk is probably write protected or missing from the drive, or there is a problem with the drive.

ERROR 0222: DOS version X.XX is already installed on your system!

The DOS version to be installed on your boot disk is already running on the user's machine.

ERROR 0223: UPGRADEOS error upgrading system files to boot disk.**ERROR 0224: UPGRADEOS: Unable to reset disk controller.****ERROR 0225: UPGRADEOS: Unable to read boot sector on boot disk.****ERROR 0226: UPGRADEOS: Unable to write boot sector to boot disk.****ERROR 0227: UPGRADEOS: Unable to read boot sector image file.****ERROR 0228: UPGRADEOS Error 'XX' opening: <FileName>**

The boot disk is probably write protected or missing from the drive, or there is a problem with the drive.

ERROR 0229: <FunctionName> returned error code: XX.

A script function returned an error in [RETV], and the report non-zero return codes option was checked for this package.

Installation Troubleshooting

If you receive any errors while installing or upgrading BrightWorks, display the log file to view the errors and possible solutions.

NOTE: Refer to the release notes for additional Install error messages.

Error calling DLL function. This indicates that install was unable to find PROGLIB.DLL or NETWARE.DRV didn't load or wasn't configured in your SYSTEM.INI file.

This could happen if the NetWare shell was not loaded before running Windows or if the wrong NetWare driver was loaded for Windows. Please refer to the installation requirements in this manual.

Also, make sure:

- The shells are loaded.

- The following line is included in your SYSTEM.INI file in the [386Enh] section: network=*vnetbios, vnetware.386, vipx.386

- You have Write and Modify rights to your Windows directory.

Install requires temporary storage on your hard drive, approximately 300K bytes. There is not enough space on your XXXX.

XXXX is the drive name specified. This message will display if the drive you specified does not have the space required to run the installation program.

Unable to copy or decompress file: FILENAME. Make sure that you have permission to write to the designated path and that you included the drive letter and that there is enough space on the destination disk.

FILENAME is the file to be copied or decompressed. This message will display if 1) you do not have the write permission 2) there is not enough space on the destination disk or 3) the volume (i.e., SYS) that the install is trying to write to does not exist. Log in as supervisor or equivalent.

Install did not find a previously installed copy of XXXXXX in YYYYYY. Choose OK to choose another path.

XXXXXX is the name of the product which you want to upgrade. YYYYYY is the name of the path you gave for the install to check for the previously installed product. This message will display if the install did not find the previously installed product which you want to upgrade in the path you specified. Make sure you give the correct path to the install to find the previously installed product for upgrading.

This installation failed. Please run the install again to be sure that BrightWorks is installed correctly. Choose OK to exit install and view the install log file.

This message will display when the installation has encountered severe problems and has aborted. A log file may have the error message. Use Windows Notepad utility to view this file. Make the required change and then run the install again.

Fatal Error: [Error #]

Verify that you meet the BrightWorks configuration requirements and then contact McAfee Technical Support with the Error #.

Install detected problems with your Configuration. Click on OK to exit and view log file.

View the Log File for information to correct your configuration in accordance with the BrightWorks installation requirements. Once you have corrected your configuration, rerun the installation process.

Unrecoverable Error

Verify that you meet the BrightWorks configuration requirements, and then contact McAfee Technical Support with the Error #.

NOTES

Index

A

About

- BrightWorks, 12
- Distribution, 47

Accessing BrightWorks, 30

B

BREQUEST.EXE, 27

BrightWorks

- Environment, 14
- Features, 12
- Icons, 32
- Installation, 17
- Menu Bar, 31
- Printing, 36
- Using the Keyboard, 34

BrightWorks Console, 30, 50

BrightWorks Icon, 30

BSTART, 27

BTRIEVE, 19, 51

BWORKS.EXE, 50

C

Configuration

- procedures, 27

D

Determining

- version numbers, 18

Distribution

- Concepts, 48
- Configuration Options, 51
- Features, 14
- Log History, 137
- Log History Details, 138
- Modules, 50
- Monitoring, 137
- Script Language, 77

Update Program, 51, 125

E

Error Messages

- Script Editing, 141
- Software Distribution, 142
- Software Distribution Update, 144

F

Fileset

- Copying, 64
- Create Directory Structure, 60
- Defined, 48
- Deleting, 64
- Directory, 56
- Editing, 61
- Features, 55
- Include Path, 60
- Renaming, 63
- Target Directory, 61, 128

H

Help, 34

I

Icons, 32

Installation

- instructions, 18
- new, 21
- upgrade, 21, 24

Installation, Troubleshooting, 148

Inventory

- Features, 13

IPX, 18

L

Log file, 20

Login Script, 53

M

Menu Bar, 31

Metering

Features, 12

Migration, 26

N

NETWARE.DRV, 18

NETX, 18

NLMs

loading, 27

O

ODI Drivers, 18

P

Package

Activating/Deactivating, 134

Advanced Options, 129

Completed, 133, 134

Creating, 126

Default Path, 128

Defined, 48

Deleting, 134

Editing, 131

Features, 125

Priority, 132

Re-distributing, 132

Renaming, 133

Start Date, 127

Status, 133

Timer Interval, 135

Update Options, 128

Packages

Priority, 126

Printing, 36

Q

Query

Applying to Scope, 116

Creating, 119

Deleting, 121

Editing, 121

Example, 117

Removing from Scope, 117

R

Report, 33

S

Scope

Applying a Query, 116

Copying, 124

Defined, 48

Deleting, 124

Editing, 122

Queries, 115

Removing a Query, 117

Renaming, 123

Script

Compiling, 69

Copying, 73

Defined, 48

Deleting, 73

Editing, 71

Entering Commands, 67, 72

Features, 65

Language, 77

Paste Script Function, 68

Renaming, 72

Script Functions, 78

ADDDEVICE, 89

ADDGROUP, 51, 95

ADDITEM, 51, 97

ADDLINE, 89

ADDPATH, 90

APPENDPATH, 100

ASSIGN, 101

ATTRIB, 83

CFGGETSTRING, 92

CFGGETVALUE, 91

CFGSETSTRING, 92

CFGSETVALUE, 91

COPY, 84

DEFINE, 82, 101

DELETEDIR, 84

DELETEFILE, 85

DOS Functions, 82

Easy System File Functions, 88

EXIT, 102

FINDFILE, 85

- GETINIINT, 98
- GETINISTR, 98
- IF-ELSE-ENDIF, 103
- MDIR, 86
- Miscellaneous Functions, 100
- NUMTOSTR, 103
- PAUSE, 104
- REBOOT, 104
- RENAME, 86
- REPLACEKEY, 92
- REPLACELINE, 93
- REPLACELINEADD, 93
- Rules, 107
- SCHEDULEWIN, 51, 99
- SETSYSFILE, 94
- SHELL, 104
- STRCAT, 105
- STRCOMPARE, 106, 131
- STRCOPY, 106
- System Variables, 109
- Types of, 78
- UPGRADEOS, 87
- Windows System File Functions, 95
- WRITEINIINT, 99
- WRITEINISTR, 99
- WRITELN, 107

Script Functions

- FINDFILE, 131

Script Language, 77

- Rules, 107

- Syntax, 77
- System Variables, 109

SDUPDATE

- Executing, 53

SDUPDATE.EXE, 51, 125, 144

- Syntax, 51

SETUP, 19

Sites

- Local Site, 47

Software Metering

- Features, 12

T

Tool Bar, 32

Troubleshooting Installation, 148

U

USAGE.EXE, 27

V

Version Numbers

- determining, 18

VIPX, 18

VNETWARE.386, 18

W

Windows Terms, 29

WSDUPD.EXE, 51, 95

WSDUPD.INI, 95