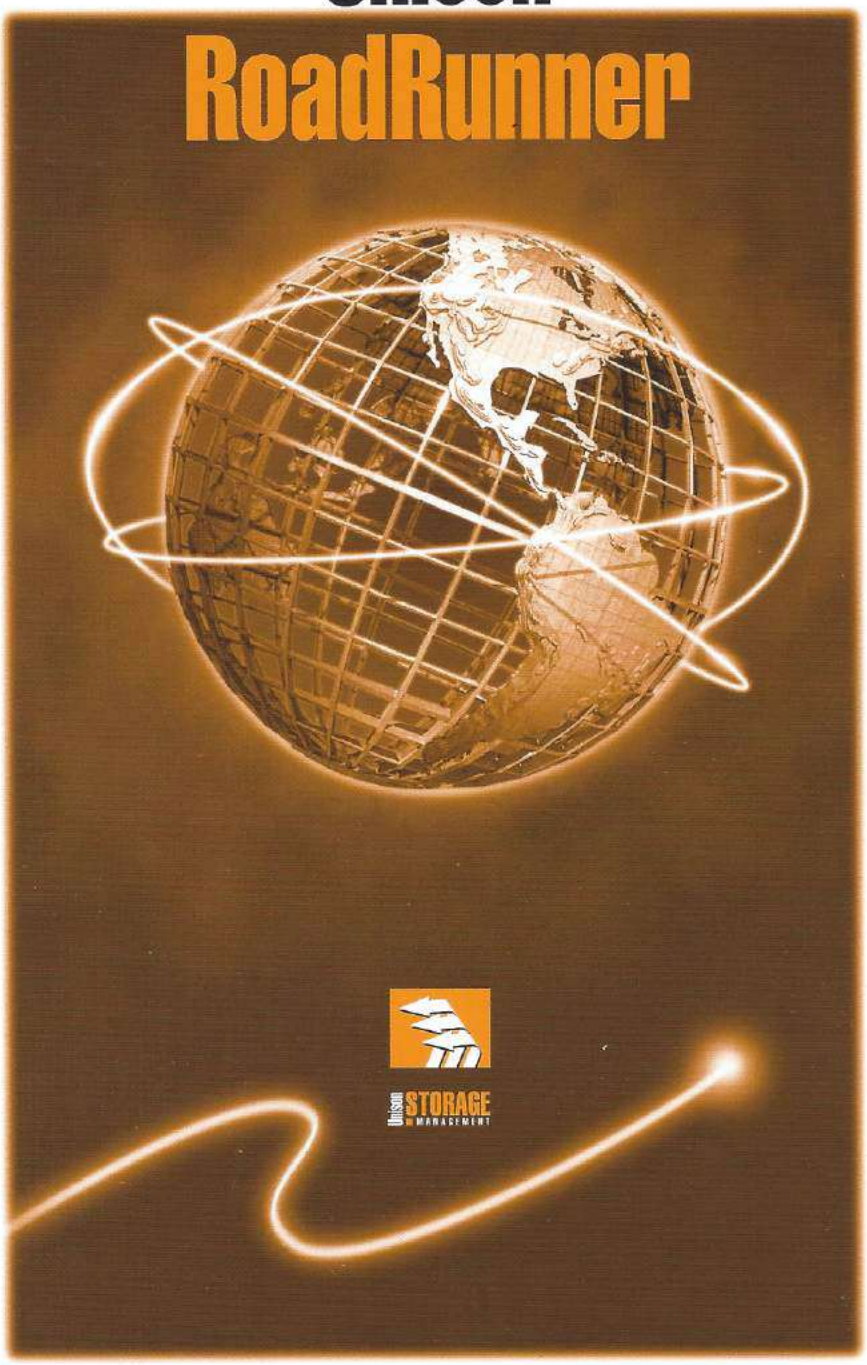
Unison

RoadRunner



Unison
SOFTWARE

Unison RoadRunner for MPE User Guide

Headquarters ■ Santa Clara, CA ■ 408/998-2800 ■ Fax 408/988-2236
Austin Division ■ Austin, TX ■ 512/478-0611 ■ Fax 512/479-0735
UK Division ■ Harpenden, Herts ■ 0582 462424 ■ Fax 0582 462425

RR-510000-1196-0

Notice

Document Number RR-510000-1196-0

Printed in USA November 96

Unison Software
811 Barton Springs Road
Austin, Texas 78704 USA
(512) 478-0611
Fax (512) 479-0735

Printing History

An **edition** of the manual represents a complete revision, in which all pages are replaced. An **update** contains additional and replacement pages to be inserted by the customer. Editions and updates are identified by a date. The date in the notice section above corresponds to that of the current edition or update. The software **version** corresponding to each edition or update is listed below; be aware, however, that not all version changes require changes to the manual.

10/92	Edition Version 4.0
05/93	Edition Version 4.1
06/95	Edition Version 4.5
05/96	Edition Version 4.6
11/96	Edition Version 4.6.5

Conventions Used in this Manual

MPE and RoadRunner commands and keywords appear in the Courier monospaced typeface, in both text and examples. MPE commands are preceded by a colon.

```
full
```
or
```
:run rr.pub.tym
```

In examples of RoadRunner syntax, the bold font designates words and symbols that you must type as shown. Italic Courier indicates a variable parameter for which you must substitute an appropriate value. Text or symbols inside brackets ([]) are optional. If brackets are nested, you can specify parameters in the inner brackets *only* if you have specified parameters in the outer brackets. In the following command:

```
report [([fields] [with options] [to devs] [format formats])]
```

- Everything except the **report** keyword is optional.
- If you use the options, they must be typed within parentheses.
- **with**, **to** and **format** must be typed exactly as shown, and each has a variable parameter for which you supply a value.

For example:

```
report (with command to 25 format 2 up, duplex)
```

Options that are mutually exclusive (you must select only one of them) are separated by vertical bars. In this example, you can choose either **ansi** or **rr**.

```
label [ansi|rr]
```

Elements that can be repeated are followed by an ellipsis (...). In the next example, you can specify as many filesets as you like; each one must be preceded by a comma.

```
excluding fileset [,fileset]...
```

For example:

```
excluding @.@.SYS,@.JUNK.MYACCT, @.@.TELESUP
```

In examples of computer/user dialogue and in examples that show both prompts and responses, user input appears in boldface.

```
ANY CHANGES?YES
or
:run rr.pub.tym
<01>select @.steve.rr
<02>store to *t
<rr>/go
```

Online Module

Descriptions of features not included in the RoadRunner base product are marked with a block in the margin, as shown at left. These modules must be purchased separately.

For specifying filesets, RoadRunner supports the following wild card characters:

`@`	Zero or more alphanumeric characters.
`?`	Exactly one alphanumeric character.
`$`	Exactly one alpha character.
`#`	Exactly one numeric character.
`[abc]`	Exactly one character from the list in brackets. A range of characters can be specified with a dash, as in `[a-z]`.

How To Use This Book

The information in this book is organized as follows:

Chapter 1, **Introducing RoadRunner**, provides a high-level description of RoadRunner's special capabilities and features. It provides cross-references to more detailed discussions of each topic. You will also find a discussion of variables affecting RoadRunner's performance on your system.

Chapter 2, **Basic Operations,** will get you started with RoadRunner. It includes a discussion of hardware and software compatibility and instructions for installing RoadRunner. It covers all basic backup tasks, among them full, partial, incremental and interim backup; directory backup; and reload. This chapter also covers how to write RoadRunner commands and how to use the Command Editor.

Chapter 3, **Advanced Operations**, contains a variety of topics of interest to advanced users. These include how to control program messages and listings, how to use the event/action mechanism, instructions for network backup, and tips for power users.

Chapter 4, **Command Reference**, provides syntax, explanation, and examples for each RoadRunner command.

Chapter 5, **Keyword Reference**, provides syntax, explanation, and examples for each RoadRunner keyword.

There are four Appendices: **BackPack/XL and TurboStore Compatibility, Utilities, Technical Information,** and **Error Messages**.

In addition to this book, you should also receive *Getting Started*, which gives installation instructions and a brief overview to get you started using Unison RoadRunner for MPE.

Where To Go For Help

If you have a question which you cannot answer using the manual or the online help system, call Technical Support at 512/478-0611 (Austin) or 408/988-2800 (Santa Clara) between 8:30 and 5:30 CST and PST respectively. In the U.K., please call Unison Software (UK) Limited at 1-582-462424. Elsewhere, call your local distributor.

When calling, please follow these guidelines:

- Know what version of the product you are using (run RR.PUB.TYM).
- If you prefer to send a fax describing the problem prior to contacting support, our fax number in Austin is 512/479-0735 and 408/988-2236 in Santa Clara. In the U.K. it is 1-582-462425.
- Technical Support can also be contacted electronically through Internet e-mail at `support@unison.com` or the Internet using `ftp.unison.com`, or `www.unison.com`.
- If you have an emergency after business hours, in the U.S, call 512/478-0611 and leave a voice mail in the emergency support mailbox. Your call will be returned within 30 minutes. In the U.K. call 01-582-767556.
- For Sales and marketing information, call any of the above numbers or email info@unison.com.
- For documentation changes, email doccomm@unison.com.

Contents

Chapter 1

Introducing RoadRunner

RoadRunner replaces HP'S STORE/RESTORE, TurboStore II and DBSTORE for Series 900 HP 3000s. It reduces system backup time and tape usage and provides many operational enhancements over all other HP 3000 backup utilities. These are discussed briefly in this chapter, with cross-references to further information.

This chapter also contains a discussion of variables affecting RoadRunner performance and an overview of hardware and software compatibility.

Flexible Backup Approach

Supports All Tape Devices: Reel to Reel, 8 mm, DAT

RoadRunner can backup data from local or remote systems to any tape drive supported on Series 900 systems. It takes full advantage of the high speed and capacity offered by your devices.

Unattended Backup to Disc

RoadRunner can back up your data in compressed format to "virtual tape" files dedicated for backup service. These tape-image permanent files may be located on a system volume set, a nonsystem volume set.

Because tape mounts are not required, the compressed disc capability can provide backup of local or remote systems with no operator intervention. Also, the combination of an operatorless store and a fast, random-access restore make virtual tape backup ideal for application checkpointing—storing a group of files about to be updated, so they can be quickly and easily restored to their original state, as needed.

RoadRunner provides special maintenance commands to purge, rename, and copy virtual tape files to other media, since they cannot be altered with standard MPE/XL commands.

Online Backup Concurrent with User Processing

Online Module

For sites that require minimum downtime, RoadRunner's online backup approach provides unrestricted read and write access to files throughout the store process with no perceptible degradation of system response. Users must close their files for a brief synchronization period at the start of the backup, then return to work with no further interruptions. If restores of selected files are required, the operator procedure is no more complex than for files backed up in a stand-alone environment.

See "Online Backup," on page 3-16.

Next-Generation Performance

Multiple Tape Devices Can be Used in Parallel or Sequentially

In sequential mode, RoadRunner begins writing to the tape on the second drive as soon as the tape on the first drive is full. This eliminates the downtime required for rewinding, dismounting, and mounting tapes. It can significantly reduce overall backup time.

Pro Module

In parallel mode (Pro Module only), multiple tape devices are written to concurrently, which cuts backup time dramatically—up to a factor equal to the number of drives in use.

Either approach may provide unattended backup, especially if high-capacity DAT drives are used in combination with RoadRunner's high compression option. From an operations standpoint, RoadRunner minimizes file spread across the media volumes in use to make restores fast and easy. Restore operations can also be performed sequentially or in parallel from multiple drives.

See "Using Multiple Tape Drives," on page 3-20.

Fast Restore and Reload

Restore performance is critical both for routine operations and disaster recovery. RoadRunner's restore operations are the fastest available in any HP 3000 product—in many cases, literally as fast as store.

When restoring from a DAT, RoadRunner takes advantage of both the high capacity of the tape and search speed of the drive. Since you can keep appending partial backups to the full backup on a single cartridge, you may only have one cartridge to mount when you need to find the current copy of a file, even though the file might have been updated daily and have been stored in multiple partials.

The maximum time to locate a file on a DAT tape using RoadRunner is normally about 40 seconds. You can usually locate a file on 8-mm tape in less than 35 minutes; other products may take up to 3-1/2 hours for a similar search.

Configurable Compression

Pro Module

RoadRunner lets you select regular compression (approximately 2:1), high compression (approximately 4:1; Pro Module only), or no compression. This gives you control over the balance between media usage and elapsed time, and helps to facilitate unattended backup. You can apply compression levels either globally or locally to specified filesets.

See "Configuring Compression:Performance Tuning," on page 3-25.

Data Interleaving

RoadRunner stores multiple files concurrently, interleaving data from the files as it writes to the backup medium. If system performance is limited by disc I/O throughput, this approach dramatically reduces backup time. You can use the **interleave** keyword to optimize backup performance for your configuration.

See the **interleave** keyword on page 5-40.

Backup Management Features

Simplified Routine Backups

RoadRunner provides one-word commands for the most common backup commands. It automatically sets and retrieves the last backup date, as required, and provides detailed prompts to simplify reload procedures. RoadRunner also supports an innovative three-tiered backup approach that dramatically cuts weekly backup by adding a monthly or quarterly backup for static files.

See "Routine Backups," on page 2-2.

Sophisticated File Selection

RoadRunner can select or exclude files from an operation based on virtually any file attribute or group of attributes, including creation, modification and access dates, file type, file size, file code, and much more. You can select or override backup options for individual filesets.

See the **select** keyword on page 5-60; the **excluding** keyword on page 5-33; and the **where** keyword on page 5-78.

Ability to Append to DAT Tapes

With RoadRunner's high density compression, you can back up as much as 8 gigabytes of data to a DAT cartridge. RoadRunner further utilizes DAT capacity by storing more than one backup to the same DAT cartridge. You can put a full backup and the rest of the week's incrementals on a single cartridge or multiple cartridges if necessary.

See "Appending Multiple Backups on DAT Cartridges," on page 2-13.

Automated Device Handling

RoadRunner's device handling keywords facilitate unattended backup by letting you configure a tape drive for auto-reply and put it online automatically. RoadRunner can also eject a DAT cartridge automatically after using it, preventing it from being accidentally written to by a subsequent operation.

See the **autoreply** keyword on page 5-8; the **autoload** keyword on page 5-6; and the **autoeject** keyword on page 5-6.

Powerful Error Recovery

RoadRunner's error recovery feature can continue a store after a write error or power failure without your having to abort and restart the operation. To control tape quality, you can specify a soft error threshold, after which RoadRunner rejects the current tape and invokes the recovery procedures used with unrecoverable errors.

On a restore, RoadRunner can restore files from a tapeset, even if the directory is damaged. If some of the reels of a tape set have been misplaced, you can restore only the reels you have. If you choose, RoadRunner can attempt to restore available data from damaged files.

See the **maxerrs** keyword on page 5-44; and the **recover** keyword on page 5-54.

Built-in Validation

As magnetic tape can be prone to media errors, RoadRunner provides the ability to thoroughly verify backup tape sets after they are created, either in their entirety or selected filesets. RoadRunner not only reads the tape block by block to detect media problems, but also checks each block for correct format and overall restorability, following the pointers through each file as if it were performing an actual restore.

See "Validation," on page 2-17.

Control Over Program Messages

You can choose which types of messages RoadRunner displays: basic messages, error messages, warnings, progress messages, statistics, or any combinations of the above. Messages can be routed to disc files or to other sessions.

See "Program Messages," on page 3-9.

Control Over Listing Format

You specify exactly what information you want printed about each file, choosing from a complete set of file characteristics. Formatting options for LaserJet-compatible printers include the ability to print reduced listings two-up or four-up on a page, duplex printing, and portrait or landscape orientation.

See "Listing Formats," on page 3-21, and the **report** keyword on page 5-55.

Pro Module

Support for ANSI and RoadRunner Format Tape Labels

Labeled tapes provide overwrite protection by allowing you to specify an expiration date before which the tape should not be used. Tape labels are often used in conjunction with a tape library management facility, such as Tapes Plus from Unison Software. RoadRunner supports both the ANSI format (Pro Module only) and a proprietary RoadRunner format.

See "Labeled Tapes," on page 3-13.

Restore on any System

For disaster recovery, RoadRunner places a copy of its restore module in MPE XL STORE format at the front of every backup tapeset. This allows you to restore files on any MPE XL system.

See "Database Backup and Restore," on page 2-7.

Ability to Print Directory of an Existing Backup

You can specify file selection criteria and formatting options to list all or some selected portions of the files on a backup volume set, whether tape or virtual tape, to tailor the directory listing to your needs.

See the **listdir** command on page 4-9.

Database Backup Features

Compatible with DBRECOV

RoadRunner can automatically store databases as part of normal system backup. It resets the dirty bit and updates the date/time stamp in the root file for compatibility with transaction logging and recovery.

Integrity Control During Restore

To preserve the integrity of your databases, RoadRunner can verify that a database was stored using RoadRunner's **dbstore** keyword before restoring it. It can also prevent users from accessing datasets until all the sets in a particular database have been restored.

Ultra-Fast Database Backup

To save time when backing up database files, RoadRunner can skip the check it usually performs to see if there is data above the high water mark in a data base. This could dramatically improve performance on some databases.

See "Directory Backup and Restore," on page 2-6.

System Management Features

Sophisticated Post-Store Processing

RoadRunner can reset the store bits of files as soon as they have been copied to the backup medium, so the files are available for access by users or job streams. To free up disc space, RoadRunner can automatically purge files after archiving them to tape. RoadRunner's event/action mechanism allows you to execute an MPE/XL command or command file after a particular fileset has been processed, after an error occurs, or after a particular volume has been completed. For example, you might want to stream a specific job or send a message to certain users as soon as a particular account has been stored. Full access to MPE/iX system variables increases the power of this feature.

See "Event/Action Mechanism," on page 3-11, the **unlock** keyword on page 5-73, and the **purge/nopurge** keyword on page 5-52.

System Directory Backup and Restore

RoadRunner can store and restore the directory structure for the entire system or for selected filesets only. It can include directory information for only the files stored on a particular tape; this means that if you send the tape to another site, it can be loaded with all the appropriate accounting structure, users, and access privileges in one step.

See "Directory Backup and Restore," on page 2-6; and "Reload," on page 4-14.

User Logoff Prior to Backup

RoadRunner allows you to automatically log off users and/or jobs in preparation for a backup.

See "XLOGOFF," on page B-2.

Configurable Execution Priority

RoadRunner allows you to specify the priority at which its operations are executed in order to minimize the time required for operations or to reduce impact on system performance.

See the **priority** keyword on page 5-51.

Support for MPE XL Volume Management Facilities

On a store operation, you can backup only files and directories on specified volume sets. On a restore, you can place files on specific volumes, volume classes, and volume sets.

See "Volume Management," on page 3-19.

Date Control on Restore

RoadRunner lets you retain the original creation, modification, and access dates or change them to the date and time of the restore.

See the **olddate/newdate** keywords on page 5-48.

Purge Control on Restore

When you restore a file with the same name as one that already exists on disc, you can specify whether RoadRunner replaces the existing file, keeps the existing file, or keeps the existing file only if it is newer. When overwriting, you can purge the existing file before the restore if disc space is a problem. If disc space is not a problem, you can restore the file to disc and then purge the disc file to ensure that the restore is successful and a usable version exists on disc.

See the **keep/nokeep/keepnew** keywords on page 5-41, and the **purgebefore/purgeafter** keywords on page 5-53.

Group, Account, and Creator Control

During a restore, you can create groups, accounts and creators if they do not exist. You can also move files into your own group and account. On both store and restore, you can change the names of multiple accounts, groups, users or files during a single store or restore. So, for example, you can duplicate three accounts for testing purposes in a single pass.

See the **create** keyword on page 5-20, the **local** keyword on page 5-42, and the **change** keyword on page 5-10.

Support for Store and Restore of ACDs

RoadRunner preserves the MPE XL Access Control Definitions (ACDs) specified for each file.

See the **copyacd/nocopyacd** keywords on page 5-19.

Command Input Features

Built-in Command Editor

RoadRunner's built-in command editor lets you create, modify, execute, and save complex multi-line commands. It can read in the contents of any text file, execute MPE XL commands in real time, and perform pre-execution syntax checking. It has a full online help facility.

See chapter 2, "Basic Operations."

Note

Read chapter 2, "Basic Operations," before attempting to create or modify RoadRunner commands. This chapter contains important information, including a detailed description of the **check** utility function.

Job Stream-Like Command Files

You can nest RoadRunner command files up to ten levels deep. Like job streams, these files can contain any programmatically executable MPE XL command, including file equations. It also supports MPE XL's CI variable replacement facility.

See the **include** keyword on page 5-38, and the **mpe** keyword on page 5-45.

BackPack/STORE Compatibility Mode

For upgrade compatibility, RoadRunner can read, translate, and execute commands intended for BackPack, HP STORE, and TurboStore II.

See appendix A.

Performance Variables

The degree of performance improvement RoadRunner provides varies from system to system. RoadRunner may reduce backup time on a system with an optimum profile by as much as 75% compared to HP-supplied utilities, but may provide little improvement on a non-optimum system. Some of the factors that influence RoadRunner's performance include:

- **The "balance" between the power of the CPU and the speed of the tape drive.** For example, when a powerful CPU is coupled with a relatively slow tape drive, such as a DAT drive, the tape drive is the limiting factor. RoadRunner operates at peak efficiency by harnessing unused CPU power to compress data and place it in a buffer,

where it remains until it is written to tape. Meanwhile, the tape drive keeps running at its rated speed.

- **Placement and size of extents.** RoadRunner spends less time searching for data when a file occupies contiguous extents on disc or when a file is spread evenly over multiple drives. It can also read a file with a small number of large extents more efficiently than one with a large number of small extents. However, RoadRunner's sophisticated disc I/O procedures minimize this problem.
- **Types of files and the nature of the data in the files.** For example, the greatest degree of data compression with the least CPU overhead is typically achieved with TurboIMAGE datasets. Text files, on the other hand, require more CPU overhead to achieve maximum compression.
- **File size.** The CPU overhead required to open and close many small files decreases the efficiency of the data compression process. With large files, the compression and storage processes experience fewer interruptions.
- **The number of device adaptors and channel adaptors and the manner in which disc and tape drives are connected to them.** If too many devices share a single adaptor (and especially if your tape drive and one or more disc drives share an adaptor), bottlenecks can occur.

Because so many factors influence RoadRunner's performance, the best way to determine your actual time savings is to try it on your system.

Chapter 2

Basic Operations

Hardware and Software Compatibility

RoadRunner runs on all HP 3000 Precision Architecture CPUs with MPE XL Version 4.x and MPE/iX 5.x, including MPE/iX 5.5.

RoadRunner can write stored data to:

- Reel-to-reel tape
- 8-millimeter tape
  (If you are running on an IEM 8-millimeter drive, use physical file marks; if you are using the Bering drive, select slow file marks. See your manual for details.)
- DAT tape
  Although DDS and audio DAT 4-millimeter tapes appear to be identical, audio DAT tapes are **not** designed to withstand the constant rewinds and repositioning required of backup media. DDS tapes also undergo stringent quality assurance tests to provide a dependable media for your backups, while audio DAT tapes do not. For this reason, Hewlett Packard and Unison Software recommend that you use only tapes manufactured to the DDS standard. For more information, see *"Success with DDS Media,"* Hewlett-Packard part number C1500-90911.
- Storage Tek 3480 devices
- Virtual tape files on disc
- If you use any third-party device, check with the vendor about RoadRunner compatibility issues.

Dual Density Tape Drives

RoadRunner provides intelligent default operations on dual density tape drives (the HP 7980, 7978, and 7974). On the 7980 and the 7978 drives, RoadRunner always performs store operations at 6250 BPI, unless you override this default by specifying `DEN=1600` on a file equation for the tape file. For the 7974, the default is 1600 BPI. If you have the special option that allows the drive to operate at 800 BPI, you can use a file equation to specify `DEN=800`.

Mountable Volumes

RoadRunner provides full support for store and restore of files that reside in groups spanned to mountable volume sets. As long as the mountable volumes are physically mounted, RoadRunner automatically performs any logical mounts and directory bindings necessary.

To use mountable volumes, ensure that you have enabled the mountable volume facility and have physically mounted the required volumes. If automatic mounting is not your site default, it is recommended that you enable it prior to the store or restore to eliminate

the need for the operator to allocate each volume set as it is requested. To enable automatic mounting, use the following console command:

```
:VMOUNT ON, AUTO
```

Routine Backups

In defining a backup schedule, most system managers aim for the following:

- To maintain a backup copy of all system data that is as up-to-date as possible.
- To minimize the system downtime required for backups.
- In the event that a reload of the whole system or of individual accounts is required, to ensure that the restore procedure will not be overly complex or lengthy.

To achieve these aims, accepted data center management practice is to perform a backup of all data on the system (full backup) once a week and a partial backup of only those files that have been modified since the full backup on a daily basis. Alternately, the daily backup may include only those files modified that day, or since the last partial backup. This is called an incremental backup.

RoadRunner provides "macros" or simplified commands that correspond to these common backup commands. RoadRunner also provides a time-saving alternative approach. All of these are discussed in this section.

Full Backup

To back up the system directory and all data files on the system, use the **full** command:

```
:run rr.pub.tym
<01>full
<02>/go
```

This initiates a complete backup of the system, including the MPE/iX and POSIX directory or directories. Since no target device is specified, this command generates a request to the system console for a logical device in device class TAPE. You must have SM or OP capability to store all files; otherwise RoadRunner stores only files to which you have access. Although IMAGE database files are stored, the dirty bit and backup date in the root file are not reset unless the **dbstore** keyword is used. See the **dbstore/nodbstore** keyword on page 5-25 for more information.

The **full** command sets the full backup date to the current system date and time, which is checked by the **partial** command.

Messages

Below is a listing and explanation of all the messages RoadRunner sends to the console during a full backup. These messages are also written to $STDLIST.

RoadRunner's first task during a store is to process your fileset specifications and build its directory of the files to be stored. After finishing the directory build, RoadRunner indicates the total number of files selected, as well as the total number of files that can not be stored. Finally, the total size of the fileset in sectors is given.

```
Starting Directory Build
Reading MPE/iX directories
Scanning volume set: MPEXL_SYSTEM_VOLUME_SET
```

```
Scanning volume set: MFG_VOLUME_SET
MPE/iX directory processing complete
Finished Directory Build
2 MPE/iX directories selected to be stored
11725 files selected to be stored
2 files rejected
5534770 sectors of data to be stored
```

RoadRunner notifies you when it begins to store files:

```
RoadRunner starting to store files to media volume 1
```

After RoadRunner has completed writing data to each media volume, it displays the following progress report:

```
RoadRunner finished writing to media volume 1 on ldev 7
11725 files (3295936 compressed sectors) written to this volume
11725 of 11725 files written so far
Last file written to this volume: Z02Z346A.PUB.SYS
```

At the end of the backup, RoadRunner displays the names of files that could not be stored:

```
FILENAME.GROUP .ACCOUNT NOT STORED BECAUSE

DSTIDD   .PUB      .SYS       File is open for write access
DSTLGTAB.PUB       .SYS       File is open for write access
```

RoadRunner then prints a summary that includes the percentage of uncompressed backup space required by the compressed backup. This information can be useful if you want to implement unattended backup procedures.

```
2 MPE/iX directories stored
11725 files were stored
2 files were rejected during the directory build
3265514 uncompressed file sectors were stored
Compressed data required 59% of uncompressed file space
```

Once the backup is complete, RoadRunner resets the backup date to the current system date and time and prints the following statistics:

```
The full backup date has been set to 08/04/92 at 13:08
Process 0:0 CPU time  : 30656 milliseconds
Process 1:1 CPU time  : 95666 milliseconds
Process 2:2 CPU time  : 78161 milliseconds
Process 3:3 CPU time  : 64223 milliseconds
Total CPU time        : 268.7 seconds
Total elapsed time    : 981.5 seconds
Average input rate    : 509 KB per CPU second
                      : 292 KB per elapsed second
Average output rate   : 172KB per CPU second
```

Partial Backup

To store all files modified since the last full backup, use the **partial** command:

```
:run rr.pub.tym
<01>partial
<02>/go
```

This backs up all data files and directories modified since the last full backup. Since no source device is specified, this command generates a request to the system console for a logical device in device class TAPE. The **partial** command sets the partial backup date to the current system date and time, which is checked by subsequent interim backups.

Program messages are essentially the same as during full backup, except that RoadRunner also prints the date of the last full backup. All files modified since then are stored.

A Three-Tiered Approach

In addition to the full/partial model, RoadRunner also supports a three-tiered approach. It reduces the amount of data stored in weekly backups, which reduces downtime and may even allow you to run unattended (for example, by cutting the amount of media required to a single DAT cartridge).

Most systems have a significant amount of data on them that doesn't change. For example, files in PUB.SYS and TELESUP.SYS, as well as application programs and other files supplied by third-party vendors, are all relatively static. Recognizing this fact, RoadRunner offers a three-tiered backup approach. You do a quarterly or monthly backup of all data on your system. In your weekly backup, you only store the files that have been modified since the quarterly. Your daily backups continue as usual.

To use this approach:

1. Schedule a backup using the **full** command once a quarter or once a month.
2. Once a week, use the **partial** command to store all files modified since the last full.
3. On a daily basis, use either the **interim** command or the **incremental** command to store all files modified since the date of last partial backup. **interim** is cumulative; it stores all files modified since the last partial each time you use it. **incremental** is not cumulative; it stores only those files modified since the last incremental or interim.

```
:run rr.pub.tym
<01>interim
<02>/go
or
<01>incremental
<02>/go
```

A minor disadvantage of this approach compared to the two-tiered models is that you have more tapes to mount if an unscheduled reload is required. You need your interims and/or incrementals, your last partial, and your last full. However, for many sites, the weekly time savings outweigh the extra time in the rare case that recovery is required. (When you perform a scheduled reload, you should always do a full backup before you bring the system down. This saves time no matter what backup approach you use.)

Setting Backup Date Flags

RoadRunner's backup macros select files according to whether their modification date is greater than the backup date flag in the bpfbdate.pub.sys file. This means that for each level of backup, there is a corresponding date flag in the bpfbdate.pub.sys file that is checked and reset to the current system date and time.

For example, to perform a full backup, RoadRunner backs up all files and resets all backup dates in the bpfbdate.pub.sys file. To perform a partial backup, RoadRunner checks the full backup date flag, selects all files modified since that date, and resets the

partial date flag to the current system date and time. The following table shows each type of backup with the date flags it checks and sets.

Backup Type	Checks	Sets
`full`	No dates	Full, partial, and interim backup date
`partial`	Full backup date	Partial and interim backup date
`interim`	Partial backup date	Interim backup date
`incremental`	Interim backup date	Interim backup date

The **datefile** keyword (page 5-22) tells these macros to use an alternate file for backup date flags. This allows you to keep multiple schedules automatically. For instance, some sites can't perform a weekly full backup all at once because of time constraints, so they stagger full backups of selected accounts or applications throughout the week. Each of these "sub-backups" could have its backup dates stored in a separate date file, thereby automating a potentially complicated task.

You can also set the date flag manually using the **fulldate** (page 4-5), **partdate** (page 4-12), and **interimdate** (page 4-8) commands.

Standardizing Operations with Include Files

Some sites need to augment the standard backup commands, adding file equations, special formatting for the listing, files to be excluded from the backup, etc. You can create an include file that contains all these options as shown in the following examples. This simplifies maintenance because when a change occurs you only modify one file.

A sample include file called backconf might be:

```
mpe "file t;dev=tape"
mpe "file headlist,old;acc=append"
mpe "file lj;dev=25"
excluding k######.@.@,q######.@.@
report (long format 4up, duplex to *lj)
after error "tell manager.sys Error \errno occurred"
```

On your full backup, you would stream the following job:

```
!job fullback,manager.sys;outclass=lp,2
!run rr.pub.tym
mpe "file t;dev=tape"
full to *t include backconf
/go
exit
!eoj
```

Though indirectfiles are supported by RoadRunner, they can impact performance. You can use include files to replace indirectfiles with no decrease in speed. For example, you can set up an include file to exclude a group of files with no common characteristic that can be described using the **excluding** keyword. You would create a text file listing fully-qualified file names (wild cards permitted). This is a file called nostore:

```
excluding @.@.junk, @.@.telesup, vt@.@.@, @.dump.@
notonvs testvolset
```

To exclude the files listed in nostore from your full backup:

```
!job fullback,manager.sys;outclass=lp,2
!run rr.pub.tym
mpe "file t;dev=tape"
select @.@.@
include nostore
full to *t
/go
exit
!eoj
```

Include files also provide a command similar to BackPack or STORE USE files, as they can contain the entire text of a command. For example, assume the following include file named nojunk:

```
mpe file purgelst,old;acc=append
select (@.pub.sys where credate > lastfull and creator <> manager
purge after each "echo \filename >> purgelst")
select (@.pub.sys)
select (@.@.@ excluding @.pub.sys)
full to (tape label ansi name f1) and (tape label ansi name f2)
```

You could execute this file as follows:

```
run rr.pub.tym;info="include nojunk"
```

The following explains how the include file nojunk executes:

The first selection selects the files in the pub.sys account that was created after the last full backup and the creator was not manager. These files are stored to tape and then purged. The purged file names are appended to the file purgelst. The second selection selects the files in pub.sys and stores them after the files that were purged by the first selection. The third selection selects all the files on the system except the file in pub.sys and stores them after the pub.sys files. The store is on two ANSI labeled tapes named F1 and F2, written in parallel.

Directory Backup and Restore

The system directory is backed up automatically by the **full, partial, interim,** and **incremental** commands and is restored by the **reload** command. To meet other directory backup needs, RoadRunner offers a variety of options through the **directory** keyword. These are useful in copying accounts or volume sets to other systems, creating test environments, etc. For more information, see the **directory** keyword on page 5-28.

When you make changes to the directory structure, you may want to do a directory-only backup. To store the directory structure without backing up data files, specify **directory** with no **select** keyword on a **store** operation:

```
store directory to 7
```

To rebuild the directory structure without restoring data files, specify **directory** with no **select** keyword on a **restore**.

```
restore directory from 7
```

Note

Whenever you perform a store with the **directory** keyword, make sure you also store COMMAND.PUB.SYS. This file describes your system user defined command (UDC) structure. When you restore a particular version of your system directory, you *must* restore the corresponding copy of COMMAND.PUB.SYS to avoid system errors. After the restore, log off and log on again to activate the restored UDCs.

Database Backup and Restore

To make database management easier, RoadRunner provides the **dbstore**, **dbrestore**, and **dbfast** keywords.

Storing IMAGE Databases

RoadRunner's **dbstore** keyword stores IMAGE databases in the manner required for database logging and recovery as part of normal system backup. It simplifies operations and reduces media requirements by compressing all databases and storing them on a single tape set.

For example, to do a full backup including all databases, use the **dbstore** keyword as follows:

```
full dbstore
/go
```

When you specify **dbstore**, RoadRunner updates the "date and time of last store" fields maintained in the root file of each database before storing them to tape, and on disc after the backup is completed. It also resets the dirty bit in each root file to indicate that the database has not been modified since the last store. If and when the database is restored in preparation for recovery, the DBRECOVR program knows that you are starting with a clean database.

If the root file is specified as part of the store fileset, **dbstore** instructs RoadRunner to automatically store all datasets that make up the database. If the root file cannot be stored, no data sets are stored. This guarantees data integrity in case recovery is required. For example:

```
select arbase.cust.sales, custdb.cust.sales
dbstore
store
/go
```

If arbase and custdb are the root files of two databases, RoadRunner automatically stores all datasets that make up these databases along with the root file. The date and time stamp and the dirty bit in the root files are updated as described previously.

If you specify the **date** parameter in combination with **dbstore** on a partial backup, RoadRunner automatically stores the whole database if any dataset meets the specified date criteria. For example:

```
select @.@.@ where moddate>5/1/96
dbstore
store
/go
```

So long as you do not specify the **online** keyword in combination with **dbstore**, RoadRunner prevents write access to the selected database during the backup.

Restoring IMAGE Databases

When restoring a database, specify the **dbrestore** keyword to ensure that RoadRunner restores the selected databases in a manner consistent with HP's automated transaction logging and recovery. Like the **dbstore** keyword, **dbrestore** restores all datasets if you specify the root file; it does not restore any datasets if the entire database was not stored using **dbstore**. For example:

```
select @.@.salebase restore from 7 dbrestore
```

You don't have to use DBUTIL to purge the database prior to the restore; RoadRunner replaces the existing files with the restored ones.

Fast Database Backup

During normal database backups, RoadRunner checks for data above the "high water mark" of the database file (a pointer to the highest active entry) because a corrupted delete chain can cause new entries to be placed above the high water mark. If you're sure that the database is structurally sound and that there are no active entries above the high water mark, you can save time when backing up the database by using the **dbfast** keyword to eliminate this check. For example:

```
select arbase.cust.sales, custdb.cust.sales store to tape
dbfast
```

System Installation and Reload Instructions

- To perform an install of your system, you need:
- A current SLT created with the MPE/iX SYSGEN facility
- Your most recent full backup, created either by using the **full** command or by using the **store** command to backup all files and directories.
- Your most recent partial backup if it is more recent than your latest full backup.
- Your most recent interim backup if it is more recent than your latest full or partial backup.
- All incremental backups that are more recent than your latest full, partial, or interim backup.

Once you have gathered these items, follow the five steps listed below. The steps are followed by a sample listing of the console messages RoadRunner displays and the commands you enter to perform each step of this procedure.

Note

Before you shut down your system in preparation for a reload, use the RoadRunner **validate** command to ensure the validity and restorability of all tape sets you plan to use in the reload procedure. If possible, do a full backup so you have only one tapeset from which to restore files.

Step 1: Load System

Mount your system load tape (SLT), boot your system, load MPE/iX and your I/O configuration, then log on as MANAGER.SYS. Complete instructions for performing a reload can be found in the *System Backup and Recovery Reference Manual*, Hewlett Packard Part No. 32650-90039.

Multiple System Volume Sets

If you have more than one system volume disc, make sure to run VOLUTIL after the load to add these discs to the system volume set.

Step 2: Restore RoadRunner

Use MPE/iX :RESTORE to restore the program ROADREST.PUB.SYS from the first tape of a RoadRunner store set. The ROADREST file contains the RoadRunner restore module in the standard MPE/iX store format.

To restore ROADREST.PUB.SYS, enter the following commands:

```
:file t;dev=tape
:restore *t;roadrest.pub.sys;show
```

Note

If you use ANSI labels, ROADREST is not placed on your backup tape. Restore it from a MPE STORE tape created for this purpose, restore RR.PUB.TYM from your product tape, or create an MPE STORE tape of RR.PUB.TYM before you start. Instructions are found in the section titled "Labeled Tapes," on page 3-13.

Step 3: Reload from Most Recent Backup

Once the ROADREST program is restored, run it from a nonconsole terminal to restore your directory and all data files from your *most recent backup tape set*, regardless of type. To do so, enter the following:

```
:file t;dev=tape
:file syslist;dev=lp
:run roadrest.pub.sys
<01>select (command.pub.sys nokeep)
<02>select (@.@.@ excluding command.pub.sys)
<03>reload from *t
<04>report offline
<05>display all, progress 5
<06>/go
```

This restores all data files on the tape with the following default options: **olddate**, **keep**, and **directory all**. The **olddate** option preserves the access and modification dates of the restored files as they are on the tape. **keep** saves time by preventing system files that have already been restored from being restored again. **directory all** restores the system accounting structure and UDCs.

HP Network Services

If you are using HP Network Services, configuration files in @.net.sys and nmconfig.pub.sys are created by the boot procedure. These should be replaced by the already-configured files stored on your backup tape by reloading as follows:

```
:file t;dev=tape
:file syslist;dev=lp
:run roadrest.pub.sys
<01>select(nmconfig.pub.sys,command.pub.sys,@.net.sys nokeep)
<02>select(@.@.@ excluding nmconfig.pub.sys,com-
    mand.pub.sys,@.net.sys)
<03>reload from *t
<04>report offline
<05>display all, progress 5
<06>/go
```

Step 4: Reload from Other Backups

You now use the **reload** command to restore files from other backups, moving in reverse chronological order to your most recent full backup. Since you are working backwards through time, the first version of a file you restore is always the most current. See step 3, "Reload from most recent backup" for more information. Older versions of files you have already restored do not overwrite the versions on disc because **keep** is the default for reload. The **reload** command works through all backups in reverse chronological order ending with your most recent full backup.

1. If you do incremental or interim backups, work through these tapes in reverse chronological order.
2. Reload all partials since the last full, again in reverse chronological order.
3. Finally, reload from your most recent full backup.

Note

After restoring files, some files in PUB.SYS show up on the listing as NOT RESTORED: OLD COPY EXISTS; KEEP SPECIFIED. These are the MPE/iX operating system files restored in step 1. Approximately 100 files are not restored, depending upon the particular version of MPE/iX on your system.

Distribution of Files

To determine where to place a restored file, RoadRunner checks the file label. If a volume class or logical device number (LDN) was specified as part of the BUILD command or FOPEN intrinsic that created the file, RoadRunner attempts to place the file in that class or on that device. If there is not enough space, RoadRunner attempts to spread the file across volume class DISC within the current volume set. If this is not possible, the file is listed as not restored with the reason Out of disc space.

For more information on placement of files during restore, see "Volume Management" on page 3-19.

If the disc configuration has changed since the store or the target system is not the same as the original, you can specify the **volset @** to guarantee that all files are restored (space per-

mitting). Do not restore files from the PUB.SYS or NET.SYS groups in this way, since they contain system files that must reside exclusively on LDEV 1.

Reload Example

After you load your system (SLT), log on as MANAGER.SYS as shown below.

```
MPE/iX:hello manager.sys
16:10/#S1/44/LOGON FOR: "MANAGER.SYS,PUB" ON LDEV #20.
HP3000 RELEASE:C.50.00 USER VERSION:X.40.12 WED, FEB 15, 1996
4:10PM
MPE/iX HP31900 B.79.06 (C) Hewlett-Packard 1987. All Rights
Reserved.
```

Restore ROADREST, the RoadRunner restore module. Mount the first tape of a RoadRunner tape set and enter the commands shown in boldface:

```
:17:51/10/Vol (unlabeled) mounted on LDEV# 7
:file t; dev=tape
:restore *t;roadrest.pub.sys;show
TURBOSTORE/RESTORE VERSION C.50.11 (C) 1986 HEWLETT-PACKARD CO.
WED, FEB 14, 1996, 4:20 PM
?17:52/#S1/36/LDEV# FOR "T" ON TAPE (NUM)?
```

If you are logged on at the console while restoring ROADREST, use the control-shift a key combination to enter your reply to the tape request.

```
INVOKING CMSTORE TO READ MPEv MEDIA
MPE/iX Transport Store-Restore, B.19.33 (C) Hewlett-Packard
1988.
RESTORE *T;@.@.@;SHOW;LDEV= 18
WED, FEB 14, 1996, 4:21 PM
WILL RESTORE 1 FILES; NUMBER OF FILES ON TAPE =1
FILENAME.GROUP .ACCOUNT   LDN ADDRESS REEL SECTORS CODE
ROADREST.PUB .SYS 2$0000008D 1 3584 NMPRG
FILES RESTORED: 1
```

Now you can run ROADREST from a nonconsole terminal to restore data files:

```
:file syslist;dev=lp
:file t;dev=tape
:run roadrest.pub.sys
RoadRunner 4.5.1* Copyright 1992-1996 Unison Software * Mon
14Feb96 5:53pm
```

Use the following **reload** command to restore all data files from a full backup. In this example, there have been no partial backups since the last full backup, so the reload is complete after the last full backup has been restored.

```
:file syslist;dev=lp
:file t;dev=tape
:run roadrest.pub.sys
<01>select (nmconfig.pub.sys,command.pub.sys,@.net.sys nokeep)
<02>select (@.@.@excluding nmconfig.pub.sys, command.pub.sys,
@.net.sys
<03>reload from *t
<04>report offline
<05>display all, progress 5
```

```
<06>/go
17:55/10/Vol (unlabeled) mounted on LDEV# 7
?17:55/#S1/39/IS "T" ON LDEV#7 (Y/N)?
=REPLY 39,Y
18:12/#S1/39/mount next RESTORE volume(reel# 2)on LDEV 7
18:14/7/LDEV #7 NOT READY
18:26/#S1/39/mount next RESTORE volume (reel# 3) on LDEV 7
18:28/7/LDEV #7 NOT READY
PROCESS 15 C.P.U. TIME = 691317 MILLISECONDS
18:39/#S1/39/LDEV 7 is now available again
PROCESS 14 C.P.U. TIME = 804063 MILLISECONDS
<rr>exit
PROCESS 1 C.P.U. TIME = 127971 MILLISECONDS

END OF PROGRAM
```

Step 5: Restart System

Finally, do a SHUTDOWN and restart your system to bring up subsystems such as NS and HPDESK. This step is necessary because subsystems require certain tables that MPE/iX builds only at startup.

Restoring Files on any HP 3000 - Disaster Recovery

For recovery purposes as well as for sharing data with other installations, you can restore files from a RoadRunner tape set onto an MPE/iX-based HP 3000 that does not have a copy of RoadRunner. For this purpose, a copy of the RoadRunner restore module in HP :STORE format is placed at the front of the first reel of every RoadRunner store tape set, except when creating ANSI labeled tapes. These do not have the restore module on them; see the warning below.

This modified version of the program is called ROADREST.PUB.SYS. It is not listed on the STORE listing for RoadRunner tape sets and is not found on your system unless you have previously restored it.

Note

The version of RoadRest must be the same or newer than the copy of RoadRunner that created the tape.

You can restore this file onto any Precision Architecture HP 3000 with the MPE/iX :RESTORE command, then run ROADREST to restore files from any RoadRunner store tape produced by RoadRunner onto that system.

To install a copy of RoadRunner's restore module on an MPE/iX-based HP 3000 system, follow these steps:

1. Mount the first reel of the RoadRunner tape set and enter the following commands:

```
:HELLO MANAGER.SYS
:FILE TYMTAPE;DEV=TAPE
:RESTORE *TYMTAPE;ROADREST.PUB.SYS
```

2. Once you have completed this process, the RoadRunner restore module is installed on the system and may be used to restore files from RoadRunner tape sets. Run ROADREST as shown below:

```
:RUN ROADREST.PUB.SYS
```

3. The only commands you can use with ROADREST are **reload, restore, listdir, redo, exit,** and MPE/iX commands (preceded by a colon). If you run the program with PARM=1, you can use the restore syntax of BackPack/XL, TurboStore II, or MPE RESTORE.

```
:RUN ROADREST.PUB.SYS;PARM=1
```

ROADREST *cannot be used to store files.* To install a complete copy of the product on a second CPU, you must purchase an additional license for the product.

WARNING

If you use ANSI labels, make sure you create a ROADREST tape for use during reloads and restores on other systems. Use MPE STORE as follows:

```
:HELLO MANAGER.SYS,PUB
:COPY RR.PUB.TYM,ROADREST
:FILE T;DEV=TAPE
:STORE ROADREST;*T
```

You need to store the tape carefully so it is available when you need to do an install. Be sure to redo this step each time you receive a new version of RoadRunner.

Appending Multiple Backups on DAT Cartridges

Depending on the model of DAT drive, DDS format tapes can hold multiple gigabytes of data. However, much of a DAT cartridge's capacity is often unused. This occurs for two reasons:

- Many backups, especially partials, are very small in comparison to the DAT's capacity.
- Backups that require multiple cartridges often have very little data written to the last cartridge.

RoadRunner's appended backup feature allows you to fully utilize the DAT drive's storage capacity by storing more than one backup to the same DAT cartridge. You can append up to 512 backups to form a "backup set," which may span multiple cartridges. The **append** keyword provides this function.

In addition to saving tape, the append feature helps organize backups. For instance, a backup set might consist of one full backup along with all the incrementals or partials done after it. If you only use your tape drive for backups, you can even leave the cartridge in the drive between stores. On the next full backup, you can start a new cartridge or reuse an outdated one.

To create a backup set, first perform a RoadRunner backup to DAT as you normally would. When you're ready to do your next backup, mount the last cartridge of your previous backup, and include the **append** keyword in the backup command.

For example:

```
incremental to dat append
or
partial to dat append
```

If there is not enough room left on this cartridge for the backup to complete, RoadRunner prompts you for another cartridge as usual.

When you append one backup to another, RoadRunner automatically numbers each backup in the set sequentially. You may also name the appended backups using the **backupname** keyword. This index number or name is stored in the backup header. If you have a cartridge with multiple backups on it and you store to it using **append** with no index number or name, the new backup is automatically placed after any existing backups. If you specify a backup number or name (using the **append** keyword), RoadRunner overwrites the corresponding backup with that number or name and any that come after it.

For example, if you specify **append 2**, RoadRunner overwrites any backups appended after the first. This capability can be useful when performing a series of partial backups after a full backup. Since each partial is cumulative, including backups of all the files stored in prior ones, you can specify **append 2** to overwrite the previous partial.

WARNING

When you specify an index number or name to be overwritten, RoadRunner erases that backup, and any that follow, in writing the new backup to that location. If a system error occurs during this process, both the old and new information can be lost.

Use this command for your original full backup. →

Use this command for subsequent partials, each overwriting the previous one. ↙

```
full to dat
partial to dat append 2
```

You can use the **append** keyword in conjunction with serial device pools (to tape1, tape2). When the cartridge in the first drive fills up, RoadRunner writes to the next tape. This may provide unattended backup when appending to an almost full cartridge. You cannot use the **append** keyword with parallel device paths (to tape1 and tape2).

When restoring files from an appended backup, you must specify the backup's index number or name with the **index** keyword. For instance, if you have a cartridge with four appended backups on it and want to restore files from the third, add index 3 to the command:

```
reload index 3 from 7
```

Backupname	Used with Store or Copy to name a backup being written.
Append	Used with Store or Copy to specify which backup to overwrite or to append at end of data.
Index	Used with Restore, Validate, ListDir, ListInfo, or Copy to specify which backup to read.

Tracking Appended Backups

To effectively use appended backups, you must track the index numbers assigned to each backup and the backup type (full, interim, partial, or incremental). To do this, route header information to a disc file with the **report** keyword.

On a full backup, you don't need to use **append** if it is the first time you are writing to the cartridge. You should still track the index number and backup type using the file equation and `report` statement shown in the following example. The second use of the **report** keyword tells RoadRunner to print its ordinary listing as well.

```
mpe purge headlist
mpe "file headlist; dev=disc"
full to dat
report (with header, nofiles, norejects to *headlist)
report
```

On subsequent incremental backups, use **report** to add information about the appended backups to headlist.

```
mpe "file headlist,old;acc=append"
incremental to dat append
report (with header, nofiles, norejects to *headlist)
report
```

At the end of the backup cycle (before the next full backup) you should print and purge headlist or rename it so that RoadRunner can start over with a clean slate for the next cycle. Examples follow:

```
:file jet;dev=1j
:print headlist;out=*jet
:purge headlist
```
or
```
:rename headlist, hd050793
```

Following is an example of the file produced by this process.

```
-------------------------------------------------------------
RoadRunner Native Header
-------------------------------------------------------------
Label is on a DAT tape backup
Backup created by RoadRunner 1.00 on MPE/iX in format 3
Backup originally created on 08/26/96 at 1:00
Command used to create backup: FULL <-------- A full backup
This is volume 1 of media set 1
Backup number is 1, first volume of backup is 1
Number of files on backup: 19365
Number of directories on backup: 3
Backup id: 213442954
Interleave level used: 4
-------------------------------------------------------------
RoadRunner Native Header
-------------------------------------------------------------
Label is on a DAT tape backup
Backup created by RoadRunner 1.00 on MPE/iX in format 3
Backup originally created on 08/27/96 at 1:00
Command used to create backup: INCREMENTAL <-------- An incremental backup
This is volume 1 of media set 1
```

This is the first backup of the set →

This is the second backup of the set →

```
Backup number is 2, first volume of backup is 1
Number of files on backup: 1276
Number of directories on backup: 3
Backup id: 213470232
Interleave level used: 4
-------------------------------------------------------------------
RoadRunner Native Header
-------------------------------------------------------------------
Label is on a DAT tape backup
Backup created by RoadRunner 1.00 on MPE/iX in format 3
Backup originally created on 08/28/96 at 1:00
Command used to create backup: INCREMENTAL
```

← *An incremental backup*

This is the third backup of the set →

```
This is volume 1 of media set 1
Backup number is 3, first volume of backup is 1
Number of files on backup: 1190
Number of directories on backup: 3
Backup id: 213494226
Interleave level used: 4
-------------------------------------------------------------------
RoadRunner Native Header
-------------------------------------------------------------------
Label is on a DAT tape backup
Backup created by RoadRunner 1.00 on MPE/iX in format 3
Backup originally created on 08/29/96 at 1:00
Command used to create backup: INCREMENTAL
```

← *An incremental backup*

This is the fourth backup of the set →

```
This is volume 1 of media set 1
Backup number is 4, first volume of backup is 1
Number of files on backup: 1080
Number of directories on backup: 3
Backup id: 213505366
Interleave level used: 4
```

RoadRunner also has the ability to list all backups on an appended backup with the **listdir** and **listinfo** commands (only), as in:

```
INDEX ALL
```

Tape Label Considerations

When appending to DAT cartridges with ANSI labels, note the following:

- You must specify the volume ID of the tape in use. If you do not specify the volume ID, RoadRunner prompts you whether to overwrite the tape. If you answer yes, RoadRunner overwrites the tape from the beginning, destroying all backups on the cartridge.
- If you specify an ANSI label when appending to an unlabeled cartridge, RoadRunner prompts you whether to overwrite the tape. If you answer yes, RoadRunner overwrites the tape from the beginning, destroying all backups on the cartridge.
- The expiration date in the label on the first cartridge in the set applies to the entire backup set. You must specify a date that ensures the protection of all the backups you intend to append to that set.

An ANSI label applies to the whole cartridge as a unit and cannot be used to identify individual backups. You can, however, use RoadRunner's internal label format to label backups individually, providing a machine-readable way of identifying each one. No matter

which format you use, if you overwrite the first backup on the tape, subsequent backups are irretrievable.

Store Error Recovery for Appended Backups

If an irrecoverable error is encountered during appended backup, two things may occur:

1. If a tape error occurs during a store to the first cartridge of the particular backup (meaning at least one complete backup is on the tape before the backup where the error occurred), RoadRunner attempts to locate the end of the previous backup and mark it as the end of the tape. It prompts you to mount another cartridge and restarts the store. (Contrary to usual practice, RoadRunner does not notify the ANSI tape library that the cartridge where the error occurred is bad, since that would prevent you from restoring files from the error-free backup that is already there.) When you attempt to restore files from the appended backup, information written in the backup trailer tells RoadRunner to look for the files on the next cartridge in the set.

   If a second error occurs while RoadRunner is seeking the end of the previous backup, it cannot write the backup trailer. Subsequent restores have no way of knowing that the appended backup is on the next cartridge. Thus RoadRunner will terminate at the point where the error occurred. However, if you have been tracking your appended backups using the system described previously, you should know which cartridge the backup is on and can avoid this problem.

2. If a store error occurs while writing to the second or subsequent reel of an appended backup (meaning part of the current backup is the only thing on the tape), RoadRunner recovers as usual. You are prompted to discard the current cartridge and mount another.

When Serial Devices are Used

If an error occurs while storing to a serial device pool, RoadRunner requests that a new tape be mounted on the device on which the error occurred. This prevents confusion in case the tape on the second device in the pool has been labeled (either manually or using ANSI labels) as volume 2 of the set. RoadRunner always begins the backup on the first device in the pool. During appended backups, RoadRunner handles errors occurring during the first volume of the set differently. Since a complete backup already exists on the cartridge where the error occurred, there already is a valid volume 1; starting the appended backup on volume 2 is not a problem.

Restore Error Recovery for Appended Backups

If an irrecoverable error occurs while restoring from an appended backup, restart the restore with the **recover media** option. RoadRunner reads the entire tape looking for the specified index number and file. Fast search capabilities cannot be used during this process, so it may be time-consuming.

Validation

Even if you use high-quality tapes and clean and replace them conscientiously, there is a possibility that bad spots on a tape may prevent one or more files from being restored. These errors are not always detected by the drive during the store operation. To verify that

a backup tape set is accurate and restorable, have RoadRunner read the tapes using the **validate** command:

```
validate from tape report
```

By default, the validation procedure reports the names of any files that cannot be restored and the total number files that can be restored. Adding the **report** keyword to the **validate** command generates a complete listing of files, much like that produced when the **store** was performed.

To validate just one reel of a tapeset or to validate a tape with a damaged directory, add the **recover media** keyword to the command.

If Errors are Found

If RoadRunner encounters a file that cannot be restored, it writes the fully-qualified file name and the reason for the problem to $STDLIST and/or SYSLIST. It then goes on to the next readable block on tape and searches for the start of a new file. Since more than one file can be written to a particular tape block, it is possible for RoadRunner to skip over files that are partially or completely contained within a block of tape containing a file with an error. When RoadRunner reports such an error, there are two ways you can proceed.

- Redo the store. This is not the most efficient approach, but it is simple and straightforward.
- Preserve as many files as possible from the tape set by comparing the **report** listing produced by the **validate** command with the original store listing. (If you didn't get a **report** listing during validation, rerun the validation with the **report** keyword. If you don't have the original store listing, use the **listdirectory** command to print out the tape set directory found on the last reel of the set.) Locate the problem file on your original listing and check to see if the files listed immediately following that file are included in your **validate** listing. Files not listed are not restorable. If you need backups of these files, store them again.

Problems indicated during validation may indicate a need to perform routine maintenance on your tape drive.

About Tapes and Tape Drives

Because backup tape sets are a critical resource for recovery, always use the highest quality tapes available. Due to the error tolerance of older HP tape drives, some HP 3000 sites have relied on less expensive tape brands and have been lax in cleaning and replacement. However, the new high-density tape drive models are more sensitive to errors. Furthermore, RoadRunner writes tape blocks twice as large as those used by some versions of MPE/iX :STORE. For this reason, RoadRunner is more sensitive to low quality media.

To ensure the integrity of backup tape sets, follow these guidelines:

- Always use high quality tapes for system backup.
- After a tape has been used for 20 to 25 backups, either clean or recertify it, or rotate it into less critical service.

It is very important to keep the tape drive clean. To ensure reliable operation, clean the tape drive (including the read/write head, the guide/scraper block, the capstan wheel, and the tape path) before starting any backup. On DAT drives, check the user's manual to see how often to clean the drive. Most drives require cleaning after every 24 hours of use.

Make this a standard procedure whether you use RoadRunner or MPE/iX :STORE to back up your files.

WARNING

Never open the door of the tape drive while writing to tape. Having the door open can cause errors to occur that are not detected by the MPE/iX I/O driver. The *HP 3000/XL Software Status Bulletin* documents this problem.

Tape drives, like disc drives, can get out of alignment, requiring preventive maintenance by your HP C.E. Look for the following symptoms:

- Numerous soft tape errors (see "Soft errors" below)
- Problems revealed during validation
- Poor store performance
- Aborts due to unrecoverable tape errors.

Soft Errors

RoadRunner provides a special error reporting procedure to help alert you that either the tape in use or the tape drive itself is not delivering acceptable performance. To do this, RoadRunner keeps count of the number of "soft tape errors" that occur on each reel. A soft tape error occurs when it cannot write data to tape on the first attempt, but then writes it successfully on a subsequent attempt. Each such error is recorded in the system log files as an `I/O ERROR - STATUS %000020`.

If more than 10 soft errors occur on a particular reel (you can reconfigure this number with the **maxerrs** keyword), RoadRunner reports the count along with a message advising you to check the tape and the tape drive:

```
WARNING: 11 soft tape errors encountered this reel
Tape may be marginal; drive may need cleaning/repair
```

Special Considerations: HP7978 and 7980

The HP 7978 and 7980 tape drives provide a built-in soft error handling procedure that prevents some errors from being logged to the system log files or reported by RoadRunner. Code A3 on the status panel on the front of the drive indicates these errors. Should you see this light come on during a store, make sure that you use RoadRunner's **validate** command to verify the restorability of the tape set. You should also clean the tape drive.

RoadRunner's Command Editor

Overview

You control RoadRunner's backup and restore functions with multi-line commands composed of the commands and keywords documented in chapters 4 and 5. This section covers the procedure for creating RoadRunner commands.

RoadRunner can read any ASCII text file, so you can create and modify commands using any editor. For your convenience, RoadRunner also includes a basic editor.

To use the command editor, run RoadRunner as follows:

```
:run rr.pub.tym
```

RoadRunner displays the prompt <01>, indicating that you are in command entry mode. You can begin entering commands and keywords at the prompt. After you press the return key, the line just entered is sent to a holding area called a buffer. Additional lines you type are placed in the buffer as well.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
```

To execute, save, or modify the contents of the command buffer, enter a slash in column one at any line number prompt:

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>/
<rr>
```

The slash terminates command entry mode, and the <rr> prompt is displayed. At this prompt, you can enter any one of the utility functions RoadRunner provides to manipulate the contents of the command buffer.

Utility Functions

Unlike the operational commands and keywords you enter at a line number prompt, RoadRunner executes utility functions as soon as you press the return key. When the function is complete, an <rr> prompt is displayed:

```
:run rr.pub.tym
<01>select @.steve.sales store to *t report (to *lp)
<02>/
<rr>go
— store messages print here —
<rr>
```

To return to command entry mode, you have two choices:

- Use the **new** function to erase the contents of the buffer and start over.
- Use the **add** function without a line number parameter to append to the end of the current command.

The following table lists RoadRunner's utility functions:

Function	Description
add	Returns to command entry mode at the next available line number; inserts a line at a specific place in the command; or appends a file to the end of the existing command
check	Reviews the current command for syntax errors.

Function	Description
`delete`	Removes lines from the current command.
`exit`	Terminates the execution of RoadRunner.
`go`	Executes the current command.
`help`	Accesses RoadRunner's online help.
`keep`	Saves the current command to a text file.
`list`	Displays the contents of the command buffer
`modify`	Edits lines in the command.
`new`	Clears the command buffer and returns to entry mode at line <01>.
`redo`	Edits the last utility command line.
`text`	Clears the command buffer and reads in a text file.

Abbreviating Commands and Keywords

To make entering commands easier, you can abbreviate all RoadRunner commands and keywords, as long as the abbreviation remains unique from all other RoadRunner commands and keywords. For a complete list of all RoadRunner reserved words, see Appendix D.

A Shortcut

If you type a slash at a line number prompt and then press the return key, RoadRunner prompts you for a utility function.

```
<03>/
<rr>
```

As a shortcut, you can enter the function right after the slash.

```
<03>/delete
```

If you enter **/add**, **/delete**, **/modify**, **/list**, or **/help**, RoadRunner returns to command entry mode after executing the function. In this example, the **list** function is on line <03>; after the list is complete, the <03> prompt is redisplayed.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>/list
   1 select @.steve.sales
   2 store to *t
<03>
```

Because the use of **/text**, **/go**, **/keep**, or **/check** usually indicates that you are finished with command entry, RoadRunner returns to the <rr> prompt after executing the function.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
```

```
<03>/keep back1
<rr>
```

Command and String Length

RoadRunner commands can contain up to 200 lines of 128 characters each. Nested files also have 200 line limits. If the internal command representation limit is exceeded, the following message is displayed:

```
Error parsing include file LONG2, line 17, column 31
Command overflow, try using fewer SELECTs or EXCLUDEs
```

Strings within quotes (such as those used with the **after** keyword and the **description** keyword) can be up to 256 characters in length. Strings can be broken onto multiple lines in the following manner:

```
"this is a string"
"that is broken onto two lines"
```

The preceding is equivalent to:

```
"this is a string that is broken onto two lines"
```

Optional Equal Signs

When specifying RoadRunner keywords that take a number as a parameter, such as **maxerrs 5** or **append 3**, you may use an equal sign for clarity. Both of the following are permitted:

```
partial to dat append 3
or
partial to dat append=3
```

MPE/iX Commands

To enter an MPE/iX command at any RoadRunner prompt, type a colon followed by the command. To edit the command, enter `:redo` at the prompt. Once the command has been executed and any messages returned, your current prompt is re-displayed, as shown:

```
:run rr.pub.tym
<01>:showtime
TUE, MAR 31, 1996, 10:01 AM
<01>
```

Using Parentheses in RoadRunner Commands

Parentheses are used in RoadRunner commands in three ways:

- Parentheses restrict the effect of keywords and options included in parentheses to specific filesets. In the following example, parentheses restrict the application of the **nopurge** keyword to the @.pub.myacct fileset.

  ```
  select (@.pub.myacct nopurge)
  select @.@.myacct purge
  ```

  Local keywords override global specifications for the filesets to which they apply.

- Any time a keyword is specified with more than one option or parameter, the

options—not the keyword itself—must be enclosed in parentheses. For example, the **report** keyword has a variety of options that can be specified to customize the listing. If you use just one of them, like this:

```
report long
```

no parentheses are required. If you have more than one parameter, like this:

```
report (long with command, norejects to *ljet)
```

parentheses must be used.

- Parentheses establish the order of operations among logical commands in **where** statements. For example, the following where statement indicates that the selected files are limited to those that are not ASCII files with record sizes of either 72 or 80 bytes:

```
where not (ascii and (recsize = 72 or recsize = 80))
```

Wildcard Characters in Fileset Specifications

RoadRunner supports the following wildcard characters:

`@`	Zero or more alphanumeric characters.
`?`	Exactly one alphanumeric character.
`$`	Exactly one alpha character.
`#`	Exactly one numeric character.
`[abc]`	Exactly one alphanumeric character from the list in brackets. A range of characters can be specified with a dash, as in `[a-z]` or `[0-9]`. Any set of characters in brackets counts as one wildcard character.
`%`	For use in HFS-Style pathnames. The `%` character represents zero or more directories within a file specification pathname. The `%` character must be followed by a slash. This is because the `%` character initiates a search for a directory, not a file, and the POSIX syntax requires a slash after directory names. If no directories are found at the end of the pathname, the `%/` characters select any files in that position in the pathname.

Utility Functions

add

Returns to command entry mode at the next available line number; inserts one line at a specified line number; or appends a file to the end of the existing command.

```
add [linenumber]
add filename [nolist]
```

When you first run RoadRunner, you are automatically in command entry mode at line <01>. You do not need to type **add**. However, once you terminate command entry mode by using a utility function, you can return to command entry mode at the next available line by typing **add** at the <rr> prompt.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>report (to *lp)
<04>/
<rr>check
<rr>add
<04>
```

You are returned to command entry mode at the next available line number.

To insert one line after a specified line number, enter **add** with a line number. RoadRunner creates a line at the position *after* the one you specified and displays that line for you to enter data. For example, if you enter **add 1**, Road Runner creates a blank line at position <02> of the command.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>report (to *lp)
<03>/
<rr>add 1
<02>store to *t
<rr>list
1 select @.steve.sales
2 store to *t
3 report (to *lp)
```

Line is added after the number you specify.

The new line has been inserted and subsequent lines renumbered.

To append the contents of a text file to the end of a command, enter **add** and the name of the file.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>/
<rr>add scottbak
Read 2 lines from scottbak
   2 select @.scott.sales
   3 store to *t
<rr>list
1 select @.steve.sales
2 select @.scott.sales
3 store to *t
```

The file is appended after existing lines.

To tell RoadRunner not to automatically list the contents of the appended file when adding, specify the **nolist** parameter.

```
<rr>add filename nolist
```

check

Reviews the current contents of the command buffer for syntax errors.

```
check
```

Use **check** at any time to examine the syntax of the current command buffer. If no errors are found, RoadRunner displays a message and returns to the <rr> prompt.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>report
<04>/
<rr>check
No syntax errors were found in the current command
<rr>
```

If it finds errors, **check** displays the line on which the error occurs, indicates the location of the error with a caret (^), and displays a brief description of the error.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>dore to *t
<03>/
<rr>check
dore to *t
^
Error processing command in line 2, column 1
Unknown or unexpected command or keyword: dore
<rr>
```

While **check** examines the command syntactically, it cannot ensure the completeness or functionality of the command.

delete

Removes lines from the current command buffer.

```
delete [linenumber [,linenumber]...]
delete [linenumber/linenumber]
```

If you specify **delete** with no parameter, it removes the last line in the command buffer. To remove a specific line, enter the appropriate line number.

```
delete 2
```

To remove more than one line without erasing those in between, enter the numbers of the lines you want to remove, separated by a comma. To remove lines 1 and 3, but not 2:

```
delete 1,3
```

To remove a range of lines, enter the numbers of the first and last lines, separated by a slash. To remove lines 1, 2 and 3.

```
delete 1/3
```

Once the specified lines have been removed, RoadRunner renumbers the lines in the command buffer and returns to the <rr> prompt.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>unwanted line in the command buffer
<04>report (to *lp)
<05>/
<rr>delete 3
- 3: unwanted line in the command buffer
- 1 Line deleted
<rr>list
1 select @.steve.sales
2 store to *t
3 report (to *lp)
```

The line has been deleted and subsequent lines renumbered.

exit

Terminates the execution of RoadRunner.

```
exit
```

If you exit RoadRunner without using **keep** to save the contents of the command buffer, it is lost.

go

Executes the current contents of the command buffer.

```
go
```

Prior to execution, **go** reviews the command for syntax errors, just like **check**. If there is a problem with the command, **go** notifies you in the same way.

As with all utility functions, you can either enter a slash followed by the return key to exit command entry mode, then type go at the <rr> prompt:

```
<03>/return key
<rr>go
```

Or save keystrokes by entering go after the slash on a command entry line:

```
<03>/go
```

help

Accesses RoadRunner's online help system. The system is structured like MPE's.

```
help [keyword [,option] | [error[nn] | warning nn]
```

If you type **help** by itself, RoadRunner displays a list of commands and keywords about which you can get help. If you type **help** followed by a RoadRunner keyword, RoadRunner displays information about the keyword, plus a list of options for more information (syntax, usage, examples, etc.). If you type **help error**, RoadRunner displays information about the last error that occurred. If you type **help error***nn* or **warning***nn* (the number must follow the word **error** or **warning** without a space), RoadRunner displays information about the specified error or warning.

keep

Stores the contents of the command buffer to a text file.

```
keep filename
```

The file created by this command is an ASCII text file with fixed-length records of 128 bytes.

list

Displays the contents of the command buffer.

```
list
```

The **list** function does not display utility functions or MPE/iX commands.

```
:run rr.pub.tym
<01>select @.steve.sales
<02>store to *t
<03>report (to *lp)
<04>/
<rr>delete 3
- 3:report (to *lp)
- 1 LINE DELETED
<rr>list
1 select @.steve.sales
2 store to *t
```

The list function displays the current contents of the command buffer.

modify

Lets you edit lines in the command buffer.

```
modify [linenumber [/linenumber]] [,...]
```

If you enter the **modify** function without a line-range, RoadRunner redisplays the last line in the command buffer and positions the cursor under the first letter of the line to be edited.

```
:run rr.pub.tym
<01> select @.steve.sales
<02> dtore to *t
<03>/
<rr>modify
dtore to *t
```

To select a single line to be edited, enter the number of the line to be edited after the **modify** command.

```
modify 2
```

To edit more than one line, enter the numbers of the lines you want to edit, separated by commas.

```
modify 1,3
```

To edit a range of lines, enter the first and last line numbers you want to edit, separated by a slash. To edit lines 1, 2, and 3:

```
modify 1/3
```

To position the cursor beneath the character you wish to change, use the space bar and backspace key combination. Do not use the arrow keys. Though they may appear to move the cursor, RoadRunner cannot detect their use.

Once the cursor is positioned under the first character to be changed, edit the line and press return. The changed line is presented for your approval. If you do not want to make further changes, press the return key again to return to the <rr> prompt or display the next line to be changed.

```
:run rr.pub.tym
<01> select @.scott.sales
<02> dtore to *t
<03>/
<rr>modify
dtore to *t
s
store to *t
<rr>
```

The modified line is displayed for confirmation. →

Using the space bar in the middle of a string of replacement text causes the characters above to be replaced by a space.

```
<02>dtorr to *t
<03>/
<rr>modify
dtorr to *t
s   e
s   e to *t
```

You can edit the selected line(s) using the same techniques and editing directives as the MPE/iX REDO command..

Directive	Effect

d	Delete the character above the D.
	`report (short to *lp)`
	`  dddddd`
	`report (to *lp)`

d>	Delete characters starting with the one above the D to the end of the line.
	`store to *t nolock`
	`  d>`
	`store to *t`

>d	Deletes from end of current line.
	`store to *t nolock`
	`> dddddd`
	`store to *t`

>	Appends text to end of current line.
	`report (to *lp`
	`> long,security)`
	`report (to *lp long,security)`

Directive	Effect

`i`	Insert characters ahead of the character above the `i`.
	`report (to *lp)`
	`  ilong,security`
	`report (long,security to *lp)`

`r`	Replace the character immediately above the `r` and subsequent characters.
	`select @.@.@`
	`  rscott.sales`
	`select @.scott.sales`

`>r`	Replaces characters at the end of current line.
	`store to *t nolock`
	`> ronline`
	`store to *t online`

`di`	Combines the functions of the `d` and `i` directives on the same line.
	`select @.pub.sys`
	`  dddiinv`
	`select @.pub.inv`

For a more detailed explanation of these commands, refer to the *MPE/iX Commands Reference Manual,* Hewlett-Packard Part No. 32650-60002.

To exit **modify** without making any changes, press the return key.

new

Deletes contents of the command buffer and returns to the `<01>` prompt.

`new`

This clears all data in the command buffer without executing the current command and displays line 1 of a new command for you to begin entering data.

redo

Edits the last utility command entered.

`redo`

Entering `redo` *without a colon* redisplays the last RoadRunner utility command entered and positions the cursor under the first letter for editing. If you enter **`:redo`** (with a colon), the last MPE/iX command entered is displayed.

You can edit the command using the same techniques and editing directives as the MPE/iX `REDO` command. See the table provided under **modify**.

text

Clears the command buffer and reads in a text file.

`text` *`filename`*

Text, like **add**, reads in the contents of any ASCII text file. The difference is that **add** appends to the end of the current command while **text** clears the buffer and replaces it with the contents of the file.

Comment Capability

RoadRunner includes the capability to insert comment lines when you create multi-line commands using an editor. By adding comment lines, you can add instructive information about the backup or your particular sequence of commands.

To add comments to a RoadRunner command buffer, you insert the comment between braces: {comment text}. RoadRunner ignores these comments—they are simply displayed on the job stdlist.

The following example illustrates the use of comments in a multi-line command sequence. The command sequence and comments were created using the RoadRunner editor.

Comment Example

```
<01> select @.@.sys
<02> select ( @.@.@ excluding @.@.sys)
<02> {comment - store SYS account at front of full backup}
<03> store to 7 autoreply
<04> report (to 25 format 4up, landscape)
<05> {print store listing on LaserJet using 4up format}
<06> /go
```

Chapter 3

Advanced Operations

With RoadRunner, you can back up your system without an operator to change tapes. This chapter discusses these backup methods.

Introduction

You can do an unattended backup by backing up the data to a disc file ("virtual tape backup").

The ability to perform unattended backups depends on having enough disk space to hold the data in compressed format. If you suspect you may not have enough space or if you want to minimize the space required for unattended backup, consider these approaches:

Pro Module

- Use multiple backup devices. If you have more than one tape drive, you can use the **to** keyword to perform parallel (Pro Module only) or serial backups. See the **to** keyword on page 5-69.

Pro Module

- Use a high compression value. If you have a large number of database files (or other files that yield a high compression ratio), high compression can reduce the amount of free space needed to perform an unattended backup by a factor of two. With less powerful CPUs, additional compression may result in additional elapsed time; if you are backing up at night unattended, you may not care. See the **compress** keyword on page 5-17 for details.
- Use RoadRunner's three-tiered backup option. By performing **full** backups monthly or quarterly, **partial** backups weekly, and **interim** backups daily, you may greatly reduce the fileset for weekly and daily backups. See "A Three-Tiered Approach" on page 2-4.
- Use disk mirroring. For more information, see "Disk Mirroring," on page 3-6. Also, see the **splitvs** keyword on page 5-66.

Automatic Device Handling

RoadRunner can automate tape drive operations to facilitate unattended backup.

- Configuring your tape drive for auto-reply makes it unnecessary to enter a `:REPLY` at the console before starting the write to tape. There are two ways to configure auto-reply. You can use the **autoreply** keyword to convert a tape drive or serial device to auto-reply mode before RoadRunner begins a backup, and return it to normal reply mode afterwards. See the **autoreply** keyword on page 5-8.

  Alternately, you can reconfigure devices to auto-reply mode using MPE/iX `SYSGEN`, making the drive operate in auto-reply mode at all times. See "Modifying Input/Output Configurations" in the *System Configuration Reference Manual*, HP Part No. 32650-90042.

- Though tapes are typically placed online when mounted, you can use RoadRunner's **autoload** keyword to ensure that tapes remain online during unattended backup or to put tapes back online to perform a second RoadRunner operation, such as a validate or restore. You can use the **autoload** keyword with the HP 7979 and 7980 drives as well as DAT units; it cannot be used with the HP 7974 or 7978, though these models do work with **autoreply**. See the **autoload** keyword on page 5-7.
- RoadRunner can eject a DAT cartridge automatically after using it, preventing it from being accidentally written to by a subsequent operation. See the **autoeject** keyword on page 5-6.

The following example performs unattended backup to a DAT cartridge, using the keywords discussed above to automate device handling.

```
!job jbackup,operator.sys
!file dat;dev=7
!run rr.pub.tym
!full to *dat autoreply autoload 90 autoeject
/go
exit
!eoj
```

What is the Capacity of my HP Disc Drive?

If you want to increase free disc space, it is useful to know the total storage capacity of your system. The following table shows the approximate capacities of the various HP disc drive models. By adding up the capacities of all your drives, you can find your total system capacity. Capacity in bytes divided by 256 gives the capacity in sectors.

Disc drive	Capacity in mb	Capacity in sectors
7933/35	404	1,578,125
7936	307	1,199,218
7937	571	2,230,468
7945	55	214,843
7957	82	320,312
7958	131	507,812
7959	304	1,187,500
7963 - one disc	304	1,187,500
7963 - two discs	608	2,375,000
7964 - three discs	912	3,562,500
C2200	335	1,308,500
C2201, C2202,	670	2,617,000
C2203	1340	5,234,000
IEM optical disc	325 per side	2,269,500

Estimate the number of tapes needed to mount by dividing the number of sectors you have to store to tape by the number of compressed sectors that will fit on each tape.

Type of tape	Approximate capacity in sector
2400' reel written at 6250 BPI	600,000
2400' reel written at 1650 BPI	160,000
1.3 GB capacity DAT - 60M	5,000,000
2.0 GB capacity DAT - 90M	7,700,000
4.0 GB capacity DAT - 120M	11,000,000

Virtual Tape Backup

During virtual tape backup, RoadRunner stores compressed data to one or more MPE files. Like a reel of tape, each file on disc acts as a volume of the store set; it contains a RoadRunner label, followed by compressed data files. You can create virtual tape files on traditional disc or optical disc devices. The discs used should be dedicated for backup service so you won't lose both your data and your backup if the drive fails.

If you have enough free space to hold all of your data in compressed format, you can use RoadRunner's virtual tape capability to implement dedicated disc backup. In this case, stored data remains on disc and is copied to tape only for archival purposes. If you need to restore files, the restore from disc offers a tremendous performance advantage over restore from tape.

Because no tape mounts are required, the virtual tape capability provides unattended backup of local or remote systems. Also, the combination of an operatorless store and a fast, random-access restore make virtual tape backup ideal for application checkpointing: storing a group of files about to be updated, so that they can be quickly and easily restored to their original state.

RoadRunner provides commands to copy, purge, and rename virtual tape files, since you cannot modify them with standard MPE/iX commands. See the **copy** command on page 4-2, the **vtpurge** command on page 4-24, and the **vtrename** command on page 4-25. Also, if you are a BackPack/XL user, please read about the differences between BackPack's handling of VT files and RoadRunner's handling of them on page A-5.

Naming the Virtual Tape Fileset

You can use any valid MPE/iX file name for a virtual tape fileset. RoadRunner forms the file names for the individual files by appending a two digit sequence number to the name you specify, just as IMAGE does for datasets. For example, if you specify the "root" file name `BACKUP`, RoadRunner names your files `BACKUP`, `BACKUP02`, etc. If the root file name is greater than 6 characters, RoadRunner replaces the last (or last two) characters of the file name with the sequence number (depending on whether the name has seven characters or eight).

The virtual tape fileset is created in the group and account specified with the `to` keyword; if you do not specify a group and account, it creates the fileset in your logon group. For example, if the backup job stream logs on as `OPERATOR.SYS,PUB`, and no group and

account is specified, the virtual tape files are built in PUB.SYS. The following security restrictions apply:

- A user without SM capability must have write and save access for the group where the virtual tape fileset is placed.
- If the user has SM capability, the virtual tape fileset can be created in any group and account. (This bypasses normal MPE/iX security restrictions.)

If a virtual tape file already exists with the same name, RoadRunner automatically purges it before the backup begins, unless you are in interactive mode. In this case, RoadRunner prompts whether or not to purge the conflicting fileset. If conflicting files are found after the backup is in progress, RoadRunner aborts.

Note that RoadRunner virtual tape files are created as privileged files to prevent unintentional access or tampering. For identification, the files are assigned the file code -21074. RoadRunner virtual tape files can be created as nonprivleged files by specifying the **nopriv** keyword on page 5-46.

Space Allocation

RoadRunner dynamically allocates disc space as needed during a virtual tape backup. If there is sufficient disc space available, RoadRunner builds as many files as necessary, up to 99 files. If the disc space needed to complete the backup exceeds the total amount of free space available, RoadRunner suspends operation, reports the problem, and asks whether you can free some disc space:

```
RoadRunner: Out of disc space in store file "virtual tape filename",
xxxx sectors needed
RoadRunner: Need xxxx sectors, available (Y/N)?
```

You have two options:

- Reply with an N to the console request and RoadRunner aborts
- Free some disc space by purging files and/or printing spool files. Then instruct RoadRunner to try again by replying with a Y to the console prompt.

Upon completion of the backup, RoadRunner prints a message to $STDLIST indicating the amount of space actually used for the backup (as a percentage of total uncompressed sectors). For example, if you get the following message, you know that your data compressed to 44% of its original size.

```
Compressed data required 44% of uncompressed file space
```

Sample Commands

To start a full backup to virtual tape in session mode, enter these commands:

```
:file syslist;dev=lp
:run rr.pub.tym
<01>full to (disc name vt1) report
<02>/go
<rr>exit
```

To start a deferred full backup in batch mode, submit a job stream like this:

```
!job jbackup,operator.sys
!run rr.pub.tym
full to (disc name vt1) report
/go
exit
!EOJ
```

RoadRunner stores existing virtual tape files unless they are expressly excluded from the fileset. To exclude them, use a **select** statement similar to the one shown below:

```
select @.@ where filecode<>-21074
```
(excludes RoadRunner VT files)

Messages and prompts are essentially the same as when backing up to tape.

Multiple Backup Paths

Pro Module

Just as you can perform parallel (Pro Module only) or sequential backup to a set of tape drives, it is also possible to back up to virtual tape files. This can be useful if you do not have enough free disc space on a single volume set to perform an unattended backup, but there would be enough room if the backup was spread out among volume sets.

Parallel backup to virtual tape files is also useful if you have a fast CPU that spends much of its time waiting on relatively slow disc drives. When specifying destination files for parallel backup, the files must have the same filename, but can be located in different groups. Virtual tape files may be specified by name or by a back-referenced file equation. The file name portion of each virtual tape file must be the same; it becomes the root file name.

```
full to (disc name vt.groupname1) and (disc name vt.groupname2)
report
```

You can perform serial backup to virtual tape filesets as well, with a command like the following:

```
full to (disc name vt.groupname1), (disc name vt.groupname2),
(disc name vt.groupname3)
```

When RoadRunner runs out of space in the first fileset, allocation automatically resumes with the next file in the serial path. When space in all serial filesets is exhausted, RoadRunner displays a prompt similar to the one below, indicating the number of sectors it needs to complete the backup:

```
Out of disc space for virtual tape fileset VT, 2180848 sectors needed.
RoadRunner: Need 2180848 sectors, available (Y/N)?
```

The first line is a message printed to `$STDLIST` (and to the console if run from a job) and the second line is the operator prompt.

Restoring Files

To restore files from a virtual tape fileset, whether the backup was created using single or multiple paths, specify the root file of the backup with the **from** keyword. Files created in parallel need not be restored in parallel. For example:

```
select @.@.@ restore from (disc name vt) report
```

Pro Module

For backups to parallel paths, indicating the root file restores the backup files in sequence. You can restore parallel backups in parallel by indicating the root file once for each path.

For example, to restore from a two-path parallel backup where the root file name is vt.vs1, use the following command:

```
restore from (disc name vt.groupname1) and (disc name
vt.groupname2)
```

When restoring from a vt file, do not specify **nopriv**. You must have read access to the virtual tape fileset in order to restore files from it.

Disk Mirroring

Disk mirroring allows users to maintain two exact replicas of data on separate sets of disk drives. If one disk drive becomes inoperable, data is still accessible on the other disk drive while the inoperable volume is being repaired. The mirrored disk facility also allows one set of volumes in the mirrored pair to be taken offline and backed up while users continue accessing files on the other volume of the pair. However, if any groups are purged from the system after the volume set is split, the files or directories from those groups cannot be stored.

Note

There is a small amount of downtime while the disk is taken offline. However, this downtime is usually less than five minutes.

The procedure to complete an online backup using the mirrored disk facility is detailed in the *Mirrored Disk User's Guide*, available from Hewlett-Packard, (P/N 30349-90003) in chapter three. For more information on disk mirroring see the **splitvs** keyword on page 5-66.

Note

The *Mirrored Disk User's Guide* contains TurboStore commands, which should be replaced with RoadRunner commands when using RoadRunner.

Overview of Defaults

The following tables explain the primary options in effect by default for each of the various RoadRunner commands. For more information, refer to chapters 4 and 5, the Command Reference sections.

Area	Default
file selection	If no file selection is specified, SM or OP users store or restore all files on the system or backup. AM users store or restore all files in their accounts. Other users store or restore all files in their group. When using **partial**, **incremental**, or **interim**, the appropriate date criteria are also automatically applied. Use the **select** keyword to control store and restore filesets.

Area	Default
MPE/iX directories	During **full**, **partial**, **incremental**, and **interim** backups, directories are stored automatically. During a **reload**, directories and UDCs are restored automatically. During ordinary **store** and **restore** operations, directories are not included, unless the user has SM or OP capabilities and you specify the **directory** keyword.
to and from devices	If the **to** or **from** keyword is not specified, RoadRunner generates a request for a device in class `TAPE`.
compression	Approximately two-to-one compression is performed on store operations (`compress=1`). Use the **compress** keyword to set higher compression or no compression.
interleaving	By default, RoadRunner uses an interleave level of 4, meaning that four files are read from disc during store operations. Use the **interleave** keyword to set any value from 1 to 8.
ACDs	Access Control Definitions are stored by default, and are restored if they are on the tape. Use **nocopyacd** to disable store or restore of ACDs.
writing to labeled tapes	By default, RoadRunner checks to see whether a tape is labeled (either RoadRunner or ANSI label) before it writes any data. You can disable the RoadRunner label check only by specifying **ignlabel**. This saves time when using unlabeled tapes as it prevents RoadRunner from skipping past the `ROADREST` file on volume 1 of the backup to look for a RoadRunner label. The label check on subsequent volumes takes little time.

Area	Default
messages	If the **display** keyword is not used to control messages, normal, warning, and error messages are displayed. If you choose to display progress messages and do not indicate an interval, they are displayed every minute.
listing	Whether or not the **report** keyword is specified, RoadRunner's minimum default listing shows which files were excluded from the operation and why. One exception to this is the **listdir** command, which always prints a more extensive listing, even if the **report** keyword is not specified. If you run RoadRunner in batch mode and specify **report** with no options, it prints a listing in the long format, including: - fully-qualified file name - volume restrictions - media volume on which the file was stored - uncompressed size of the file in sectors - numeric and mnemonic MPE/iX file codes - size of record in bytes - number of records in the file - blocking factor of the file - number of extents in the file. If you run RoadRunner interactively and specify **report** with no options, it prints a listing in the **short** format, including: - fully-qualified file name - volume restrictions - media volume on which the file was stored - uncompressed size of the file in sectors If you specify a list of fields to be reported, overriding the default format, RoadRunner prints the fully-qualified file name for each file, whether you request it explicitly or not. If you specify no other device, RoadRunner listings are routed to `SYSLIST`, which defaults to `$STDLIST` unless overridden by a file equation. In batch mode, `$STDLIST` is the output device designated for that job (usually the system line printer); in session mode, it is the terminal from which the operation was initiated.
access and modification dates	During a store operation, RoadRunner does not change the access dates of the files selected. When files are restored, the access and modifications dates of the files are changed to the date of the restore. During a reload, access and modification dates of restored files are preserved as they were at the time of the backup. You can override these defaults with the **newdate** and **olddate** keywords.

Area	Default
overwriting files on disc	The **restore** command overwrites existing files on disk with identically named files from the backup. The **reload** command does not overwrite existing files. You can override these defaults with the **keep**, **nokeep**, and **keepnew** keywords.
ROADREST	The ROADREST.PUB.SYS program, used to restore RoadRunner backups on systems without a copy of RoadRunner, is written at the start of all store tape sets unless you specify **norrest** or when you use an ANSI labeled tape.
creating entities	When creating accounts, groups, and users, RoadRunner assigns default capabilities to the new entities, unless you specify **create usedir**. This gives the new entities the capabilities specified in the MPE/iX directory on the backup.

Program Messages

Types of Messages

RoadRunner messages fall into a number of different categories, described in this section. An exclamation point in a message indicates that a value, such as a file name or a volume ID, is substituted in by RoadRunner at the time the message is generated.

- Basic program status messages, called **normal** messages. These are generally sent when RoadRunner begins or completes a particular aspect of an operation. Examples are:

```
Starting Directory Build
Finished Directory Build
! sectors of data to be stored
Tape may be marginal or drive may need cleaning/repair
```

- Messages produced when a minor problem occurs, called warnings:

```
Unable to open RoadRunner help file: !
The volume restriction for file ! was changed to volume set ! due to
insufficient disc space in the file's original specification
File ! purged BEFORE restore due to insufficient disc space
```

- Messages produced when a serious problem occurs that prevents RoadRunner from continuing with the operation, called errors:

```
POWERFAIL detected on ldev !
No files were found to restore
Invalid ACCOUNT name/pattern for '!' in indirectfile
Unable to allocate disc space for an internal buffer
A backup media error occurred while restoring file, since the
RECOVER option was selected, an attempt will be made to restore
as much of the file as possible.
```

- Messages that show what percent of the operation is complete, called progress messages:

```
RoadRunner Progress:
   !%% (! sectors processed so far)
   Current file being processed is !
   ! files selected so far
   ! files rejected
```

- Information on the amount of CPU time and elapsed time a particular operation has taken, and throughput rate for the operation, called stats:

```
Process !:! CPU time   : ! milliseconds-
Total CPU time       : !.! seconds
Total elapsed time   : !.! seconds
Average input rate   : ! KB per CPU second
                       ! KB per elapsed second
Average output rate  : ! KB per second
```

Routing Messages

Normal, warning, and error messages are always displayed to `$STDLIST` (in batch mode, the designated output device for a job; in session mode, the terminal). You can control additional messages with the **display** keyword, discussed on page 5-31. Some examples of the ways you can use this keyword are given below.

- To avoid monitoring the system console, you might route program messages to the terminal in your office. Assuming you are logged on as `MANAGER.SYS`:

```
full to tape
display (all to MANAGER.SYS)
```

- You can also route messages to a session number, or combine these options:

```
full to tape
display (all to #S15)
display (all to KATHY.CHICAGO, #S26)
```

- You can route messages to a disc file for permanent tracking of tape errors, the locations of various versions of files, etc. If you use append access with this file, you can add new information during every backup. For example:

```
full to tape
mpe "file backinfo"
display (normal, errors to *backinfo)
```

After building and using this file the first time, you change your file equation so that information is appended to the existing file:

```
full to tape
mpe "file backinfo,old;acc=append"
display (normal, errors to *backinfo)
```

A similar capability is provided by the **report** keyword, which gives you the ability to produce printed or disc-based listings containing basic backup information. For example, you could produce a formatted listing on a LaserJet including the backup header (date and time of backup, number of tapes used and files stored on each, etc.) and the volume IDs of the tapes used for each backup. See page 5-55.

- You can use the **display** keyword more than once to route some messages to a file and others to a user or session. The following commands send backup statistics to a file and all messages (normal, errors, warnings, progress and stats) to `$STDLIST`.

```
full to tape
mpe "file backinfo,old;access=append"
display (stats to *backinfo)
display all
```

MPE/iX Capabilities

RoadRunner provides several ways to take advantage of the MPE/iX command interface. You can execute MPE/iX commands directly from RoadRunner, either immediately or as part of a RoadRunner command. You can include MPE/iX variables in RoadRunner commands, and you can incorporate RoadRunner commands into an MPE/iX job stream.

To enter an MPE/iX command for immediate execution at any RoadRunner prompt, type a colon followed by the MPE/iX command.

```
:run rr.pub.tym
<01>:showtime
TUE, MAR 31, 1996, 10:01 AM
<01>
```

To edit and re-execute the MPE/iX command, enter `:redo` at the RoadRunner prompt. This allows you to edit the command exactly as if you were using MPE/iX's REDO function. Once RoadRunner executes the command, your current prompt is re-displayed.

To execute an MPE/iX command just prior to the RoadRunner command in which it is included, use the **mpe** keyword on page 5-45. This allows you to include file equations in RoadRunner commands and provides jobstream-like capabilities:

```
<01>select @.pub.sys
<02>mpe "file t; dev=tape"
<03>store to *t
```

To utilize MPE/iX variable substitution capabilities at any point in a RoadRunner command, see "MPE/iX variable substitution," on page 3-26. This section discusses how to use MPE/iX variables to control date- and time- sensitive options, such as what device is used or what execution priority is selected.

Event/Action Mechanism

RoadRunner can execute MPE/iX commands or command files at specified points in its operation: after storing or restoring certain files, after a certain reel is complete, after an error occurs, etc. Event processing is performed by the **after** keyword, described on page 5-1. This section provides some examples of ways this capability extends the scope of your control over the backup process.

Any processing initiated by the event/action mechanism causes RoadRunner to suspend backup operations until the requested action is complete. To avoid lengthy interruptions of the backup, stream batch jobs for any actions requiring more than a few seconds.

In the case of the **after all** and **after each** options, files are considered processed once their store bits have been reset. The setting of the **unlock** keyword affects when the specified actions are performed.

After All

You can use this option to notify a particular user when the files they need to use have been stored and are ready to use. To make the files immediately accessible, you must specify **unlock byfile** so that the store bit of each file is reset as soon as it has been copied.

```
full to tape
select (@.@.mktg after all "tell mgr.mktg MKTG acct is ready to use")
```

After Directory

Online Module

This feature is useful for informing users when they can log back on after the synchronization period required for online backup is over.

```
full to tape online
after directory "tell @ All files are ready for access"
```

After Error

In the following example, a full backup is done to three tape devices in parallel. Assuming such a program were available at your installation, the **after** keyword is used to run the program XCALL.PUB.PROGS that dials the modem and calls the system manager if an error requiring operator intervention occurs during the backup. RoadRunner's **\errno** parameter accesses the error number, which presumably would be reported by the program. Backslashes (\) identify double quotes used within the **after** string.

```
full to tape1 and tape2 and tape3
after error "run
xcall.pub.progs;info=\"\errorno\"xcall.pub.progs;info=\"\errorn
o\"
```

In the following example, RoadRunner writes the error number to a log file when an error occurs:

```
mpe "file errlist,old;acc=append"
select @.@.acct1,@.@.acct2
store to tape
after error "echo RoadRunner error number \errno occurred at
\time >> *errlist"
```

It is also possible to automatically abort RoadRunner when a tape error occurs, rather than have it perform its usual recovery procedures:

```
incremental to tape
after error quit
```

After Each

In the following example, files in the PUB.SYS account that were created since the last full backup by a user other than MANAGER.SYS are purged. After each file is purged, its name is written to a file. The RoadRunner system-defined parameter **\filename** captures the name of the file currently being processed:

```
mpe file purgelst,old;acc=append
select (@.pub.sys where credate > lastfull and creator <>
manager
purge after each "echo \filename >> purgelst")
```

```
select (@.@.@)
full to (tape label ansi name f1) and (tape label ansi name f2)
```

Any action performed repeatedly after each of a large group of files can drastically impact backup performance. Avoid using **after each** with large filesets.

After Mediaswitch, After Mediavolume

In the next example, two devices are written to serially. RoadRunner sends a message to the operator when the first two tapes are complete, saying that it is time to mount another tape. After each reel is completed, RoadRunner writes a line to a disk file indicating the ANSI volume ID and the time of creation. This is useful in managing labeled tapes, if you do not have an automated tape management facility.

```
mpe "file vollist,old;acc=append"
full to (tape label ansi name dev1), (tape name dev2)
after mediavolume 2 "tell op.sys Please mount tapes."
after mediaswitch "echo Volume \volid written at \time as reel
\volno
>> *vollist"
```

Labeled Tapes

Tape labels provide a machine-readable way to identify store tape sets and to prevent them from being unintentionally overwritten. They are also used at many installations to automate control of the tape library. RoadRunner supports two different types of tape labels:

Pro Module

- ANSI labels (Pro module only). This format ensures that the expiration date you establish to protect your stored data can be recognized by the MPE/iX file system. ANSI labels are also read by MPE/iX's Automatic Volume Recognition (AVR) facility. Whenever you mount a tape with an ANSI label and put it online, AVR sends a message to the console reporting the volume ID. This means you can address an ANSI-labeled tape by its volume ID (for example with the RoadRunner **to** or **from** keyword), and it can be located by the system, regardless of the device on which it is mounted. Furthermore, HP's Tape Management Facility tracks the creation and use of labeled tapes and provides information to third-party tape management systems, such as Tapes Plus from Unison Software.
- RoadRunner labels. This is an internal format, the same as that supported by BackPack. Like ANSI labels, RoadRunner labels provide an expiration date, but it can be read only by RoadRunner. Neither MPE/iX nor other third-party products can recognize a RoadRunner-labeled tape or respect its expiration date. Use RoadRunner labels in these cases:
  - If you have been using BackPack labels and wish to continue in the same manner.
  - If you rarely create tapes except using RoadRunner.
  - If you do not have an automated tape management facility and want to avoid the overhead of separately naming each tape volume you mount.

You can find the syntax for specifying tape labels in the descriptions of the **from** keyword on page 5-35, and the **to** keyword on page 5-69. For information about using labels with appended DAT backups, see "Tape Label Considerations," on page 2-17.

WARNING

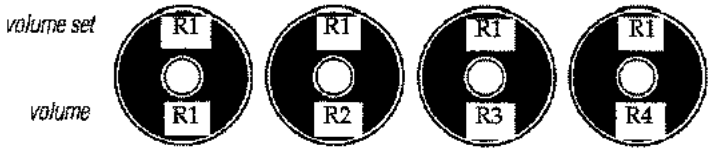
When writing to tape, RoadRunner's default is to automatically place a copy of its restore module, ROADREST, at the beginning of the backup in MPE transport format. This is used both during an install and to restore files from a RoadRunner tape at a site that does not have the product. Using ANSI labeling on your store tapes cancels this function; ROADREST is not placed on the tape. If you plan to use ANSI labels, make sure you create a ROADREST tape for use during reloads as follows:

```
:HELLO MANAGER.SYS,PUB
:COPY RR.PUB.TYM,ROADREST
:FILE T;DEV=TAPE
:STORE ROADREST;*T
```

You need do this task each time you receive a RoadRunner product update tape. Store the tape carefully so it is available when you need to do an install.

Contents of the Label

ANSI tape sets have two levels of identification: a volume set ID and individual volume IDs. On a multi-tape backup, the volume ID supplied for the first tape (either by you or by an automated tape management facility) becomes the volume set ID for that and all subsequent reels. For example, if the first tape of a four-reel backup is named R1, the volume set name for all subsequent reels is R1. In the illustration below, R1 is the volume name of the first tape in a four-volume backup.



If you have a tape management facility, it provides the volume ID for each subsequent reel automatically. If not, you can mount any existing ANSI labeled tape, and its volume ID is preserved. If you mount non-ANSI labeled tapes, you are prompted at the console to supply an ID for each one.

RoadRunner tape labels use a single volume set ID for all tapes in the set and do not have a second level of naming.

ANSI labels also contain an expiration date and a creation date. The creation date is updated each time the tape is written. RoadRunner labels contain an expiration date only.

Using ANSI Labeled Tapes

If the Tape is Already Labeled

To store to a previously-labeled tape, you must use either a file equation or the **devdescr** parameter of the RoadRunner to keyword to identify the tape. For example, to store to a volume set called VOL07, you could do either of the following:

```
mpe "file t=partial.june15;dev=tape; label=vol07,ans,12/31/96"
partial to *t
or
partial to tape name backup volid vol07 expdate 12/31/96
```

If you enter a command like those shown above and the labeled volume has not yet been mounted, the system generates the following message to the console:

```
MOUNT REEL 1 OF VOLSET VOL07 ON LEV 7
```

Alternately, if you have a tape library system that knows the volume ID of the next volume to be used and assuming, for this example, that the next volume is VOL08, the message would look like this:

```
MOUNT STORE MEDIA VOLUME VOL08 OF VOLSET VOL07 ON LEV 7
```

Once you mount the correct volume, the Automatic Volume Recognition facility (AVR) produces this message:

```
VOL07 (ANS) mounted on LDEV# 7
```

When this volume is complete, you are prompted to mount the next reel of the set:

```
Please mount store media volume 2 of set VOL07 on ldev 7
```

AVR messages confirming that the correct tape has been mounted are not produced for reels after the first.

If the Tape is not Labeled

If you are storing to tapes that are not labeled, use either a file equation or the devdescr parameter of the RoadRunner to keyword to identify the tape. For example, to store to a volume set called VOL07, do either of the following:

```
mpe "file t=partial.june15;dev=tape; label=vol07,ans,12/31/96"
partial to *t
or
partial to (tape name backup volid vol07 expdate 12/31/96)
```

The system then prompts you to mount the tape.

```
Mount volume VOL07
```

Once you mount a tape, the following message is produced:

```
Vol (unlabeled) mounted on LDEV# 7
```

You now specify the device on which the tape is mounted.

```
=REPLY pin#,7
```

The mounted volume is labeled according to your specifications. When that reel is complete, you are prompted to mount another and to provide a volume ID for this second volume, if one is not already present.

```
Reply with volume id for volume on ldev 7
Please mount store media volume 2 of set VOL07 on ldev 7
```

Reply with the volume ID for this volume:

```
=REPLY pin#,VOL07
```

This process is repeated for each reel.

If you specify a label with either a file equation or the **devdescr** parameter but fail to include a volume ID, and you have no tape management facility on your system, the tape is automatically labelled `$SCR` (meaning scratch) and you are prompted as follows:

```
Mount volume $SCR (ANS) for USERNAME
```

When Errors Occur

If one of the labels on a tape from a backup set with ANSI labels is overwritten or damaged, the tape is rejected by the Tape Management Facility. Use RoadRunner's **recover media** option on page 5-54, to recover the data on the tape.

Online Backup

Online Module

For sites that require minimum downtime, RoadRunner's online backup approach provides unrestricted read and write access to files throughout the store process with no perceptible degradation of system response. This approach can be used for storing all files, or only for selected applications and data.

How it Works

At the start of an online backup, users must exit their applications briefly so RoadRunner can establish a "synchronization point." This is the baseline for the backup, used to ensure the integrity of interdependent files and datasets. During the synchronization period, RoadRunner reads your fileset specifications and flags the files to be backed up. Users can then get back into their files as RoadRunner goes on to store data to the backup medium.

There are two online backup methodologies: rollforward recovery and rollback recovery. When files are restored, rollforward recovery restores files to their state at the end of the backup (contents are rolled forward). Rollback restores files to their state at the beginning of the backup (contents are rolled back).

RoadRunner uses the rollback approach. It works like this. As the store proceeds, RoadRunner logs write transactions to flagged files. After a particular file has been stored, it terminates logging for that file and writes the logged transactions, if any, to the backup medium immediately following the file. If you restore the file, RoadRunner reverses the logged transactions in the version it places on disc, returning the file to its state at the synchronization point.

With rollforward, files are synchronized at the conclusion of the operation. All transactions to all files are logged throughout the backup, and the transactions are written to the backup medium in a group after the data files.

Rollback provides a number of advantages over rollforward. The time to log off for synchronization can be predicted and scheduled, rather than happening whenever the backup is finished. If a system failure occurs during the backup, all files that have been stored can be restored to a consistent state, and any file modified by other users or processes during the backup can be returned to its original state. If a restore is required, you can usually find the file and its transactions together on one reel. With rollforward, all the transactions are at the end of the backup, so multiple tape mounts are often required.

Online Procedure

From an operations standpoint, online backup is not complicated. Follow the steps below.

1. **Make files available for directory build.** Files open for write access during the directory build are not stored. Have users exit their applications or log off before you start RoadRunner to ensure all files are stored in a consistent state. You can use the MPE :WARN command as shown below.

   ```
   :WARN @s Online backup begins in a few minutes. Please close
   your files.
   ```

2. **Start RoadRunner backup.** When you are sure no files are open for write access, start the backup with a command like this:

   ```
   full to tape dbstore online
   after directory "tell @ All files are now ready for access"
   ```

   When the directory build is complete, RoadRunner's **after directory** option notifies users that they can return to work.

   You can optimize performance and disc space usage as follows. Since RoadRunner stores files in the order they are selected, specify your most heavily used files, groups and accounts in a separate **select** statement preceding any others, as shown below. Once these files are stored, the extra overhead and space required to log transactions to these files is eliminated.

   ```
   select @.@.custbase,@.@.software
   select (@.@.@ excluding @.@.custbase, @.@.software)
   full to tape dbstore online
   ```

   If you have very limited disc space, you can further minimize space requirements by adding the **unlock byfile** option to your command. RoadRunner ordinarily terminates logging for a file when the media volume on which it is stored is complete. This option terminates logging as soon as the file is stored.

   ```
   full to tape online unlock byfile
   ```

3. **Respond to tape requests promptly.** Long delays in mounting tapes mean extra logging, more CPU time, and greater disc space and tape requirements. Mount tapes promptly to streamline your online backup.

4. **When the backup is complete, check your listing as usual.** If any files were open during the directory build, they are shown as not stored for the following reason:

   ```
   File was open for write access
   ```

Troubleshooting

What if the system runs out of disc space during online backup?

Some files may not be stored; these are indicated in the listing as not stored with the message: `An error occurred while processing online data`. Here is why. Transactions for files on each volume set are logged to a file on that volume set. If you run out of disc space on any volume set, logging of changes to files on that set terminates. If a file has already been stored, it is fine; if a file has not been stored but no further changes are made to it, it can still be stored without a problem; files on other volume sets are not affected. However, files on the affected volume set to which changes are made after logging terminates are not stored.

What if one of the files to be stored is renamed or purged while online backup is in progress?

The file appears on the backup medium as it was at the start of the store.

What should I do if I have a system failure during online backup?

The most reliable approach is to start the backup over.

How do I restore files from an online backup?

Use the **restore** or **reload** commands exactly as with a conventional backup. RoadRunner takes care of rolling back the file to its status at the synchronization point automatically.

How do I stop an online backup?

Use any of the methods ordinarily used to terminate RoadRunner. See "Starting Execution and Entering Commands" in the *"Getting Started Guide"*.

Network Backup

RoadRunner performs backups over a network to virtual tape (vt) files on disc or to any backup medium. For maximum performance, execute the RoadRunner backup from a job stream. Specify the target device with one of the following back-referenced file equations:

```
:file t=t:nodename;dev=devicename  (tape or vt filename)
or
:file t;dev=nodename#devicename  (tape or vt filename)
```

You must be logged on to the remote system in order to access the remote tape drive an the tape drive should be configured for autoreply. If it is not, you can add the following command after the **remote hello** in your job stream to run the RoadRunner `AUTOREP` utility.

```
!remote run autorep.pub.tym
```

The **autoeject**, **autoload** and **autoreply** keywords are not supported with remote systems.

You cannot use ANSI or RoadRunner labeled tapes when performing network backups.

Sample Jobstreams

You can use the following sample jobstreams to perform a full backup to a remote node:

```
!job rmtstor,operator.sys
!remote hello operator.sys;dsline=nodename
!file rrvt;dev=nodename#devicename
!run rr.pub.tym
full to *rrvt dbstore
report
/go
exit
!eoj
```

Performing the remote backup in a jobstream directly to the remote node (as shown above) provides a time savings over first performing a virtual tape backup to disk and

then transferring the compressed backup to the remote node. However, using the :DSCOPY command to copy your backup to the remote node as part of the same job as your local backup can save substantial CPU time, as shown in the following example:

```
!job rmtstor,operator.sys
!file rrvt;dev=devicename
!run rr.pub.tym
full to *rrvt dbstore
report
/go
exit
!remote hello operator.sys; dsline=nodename
!dscopy rrvt@;@:nodename;fcode=-21074
!eoj
```

When performing a remote backup to multiple virtual tape files, the target files must reside on the same system.

```
!job rmtstor,operator.sys
!remote hello operator.sys;dsline=nodename
!file r1=rrvt1.pga;dev=nodename#DISC
!file r2=rrvt2.pgb;dev=nodename#DISC
!run rr.pub.tym
full to *r1 and *r2 dbstore
report
buffer
/go
exit
!eoj
```

When purging virtual tape files on a remote node with the **vtpurge** command, you must use a file equation to specify the files to be purged.

```
!job rmtstor,operator.sys
!remote hello operator.sys;dsline=nodename
!file rrvt;dev=nodename#vtfilename
!run rr.pub.tym
vtpurge *rrvt
report
/go
exit
!eoj
```

Volume Management

Storing and Excluding Volume Sets

Add the **onvs volsetlist** keyword to any **store** command to specify volume sets to be copied to the backup media. For example, to back up only the system volume set and a volume set called ACCTG_VOLUME_SET:

```
select @.@.@
store to tape
onvs mpe_xl_system_volume_set,acctg_volume_set
```

Add the **notonvs volsetlist** keyword to any **store** command to specify volume sets to be excluded from the backup. For example, to exclude all files on a volume set called TEST_VOLUME_SET:

```
select @.@.@
store to tape
notonvs test_volume_set
```

Restore Basics

RoadRunner provides three keywords that control where a file is placed within the volume structure set up on your system. These are **vol**, **volclass**, and **volset**.

If you do not use any of these keywords, restored files are placed according to the restrictions in their file labels, which are those in effect when the file was stored.

If none of the keywords is used and RoadRunner is unable to use the volume restriction in the file label (perhaps because the specified volume or volume class does not exist), the file is allowed to span all volumes in the volume set to which the group containing the file is bound. No warning message is printed. Any volume or volume class restriction in the file label is changed to a volume set restriction.

When you use one or more of these keywords, the existing restriction in a file's label is replaced with the new restriction you specified, even if RoadRunner is unable to place the file in the volume, volume class, or volume set indicated. A file must *always* be restored into the volume set to which its group is bound. No matter what volume restrictions you specify or exist in the file label, the group is the primary factor determining where the file is placed when restored. Within its home volume set, the file can be placed in any volume or volume class that exists and has room for it.

The **volume** keyword is described on page 5-77. The **volclass** keyword is described on page 5-74. The **volset** keyword is described on page 5-75.

Note

Do not restore files from the SYS account with alternate volume restrictions. Certain files in the SYS account must be located on the master volume of the system volume set.

The **report** keyword provides options for printing volume information. For more information see page 5-55.

Using Multiple Tape Drives

If you have multiple tape drives, you can improve performance or provide unattended backup by specifying more than one device for your backup. All devices used for a single backup must be the same type of device—magnetic tape, DAT, etc.—but need not be in the same device classes.

Pro Module

Multiple tape drives can be specified for backup and restore operations either in sequence (as soon as RoadRunner finishes writing to the first drive, it begins writing to the second) or in parallel (writes to multiple drives simultaneously; Pro Module only). RoadRunner supports the use of up to 16 drives for parallel configurations and an unlimited number of drives in sequence. To specify drives in sequence, separate them with commas in the **from** or **to** statement. To specify drives in parallel, separate them with **and**.

```
store to *tape1, *tape2 ◄—— Stores to two drives in sequence
store to *tape1 and *tape2 ◄—— Stores to two drives in parallel
restore from *tape1, *tape2 ◄—— Restores from two drives in sequence
restore from *tape1 and *tape2 ◄—— Restores from two drives in parallel
```

You can use the parallel and sequential approaches in combination:

```
store to *tape1,*tape2 and *tape3
```

In the previous example, data is written to two parallel paths. The first path consists of two devices used serially and the second consists of a single device. The main concern in deciding how to use multiple drives is CPU power. If you only have enough CPU to drive two paths in parallel, you should use additional drives serially within the two main paths.

Example

The following job stream performs an unattended backup with multiple tape drives. The **autoreply** and **autoload** keywords are used to automate tape drive operations. Tapes are placed back online after backup to be validated:

```
!job jbackup,operator.sys
!file t1;dev=7
!file t2;dev=8
!run rr.pub.tym
full to *t1, *t2 autoreply
/go
validate from *t1,*t2 autoload 120 autoreply
/go
exit
!eoj
```

Error Recovery

When a tape error occurs during serial backup, RoadRunner prompts the operator to mount a new reel on the same drive where the error occurred. This ensures that, for example, the second reel is always created on the second device specified with the **to** keyword.

Copying Backups Created in Parallel

Regardless of the number of tape drives used to create a parallel backup, only two tape drives are needed to duplicate all or part of the backup with the **copy** command. RoadRunner preserves the sets that comprise the backup, copying files from set 1 to set 1, files from set 2 to set 2, etc. However, if you are only copying a few files from each set, or when you are copying from a low-capacity reel-to-reel tape to a high-capacity DAT, this is inefficient in terms of media usage. The **combine** keyword merges the source sets and outputs a combined destination set, or multiple sets. See page 5-16 for examples.

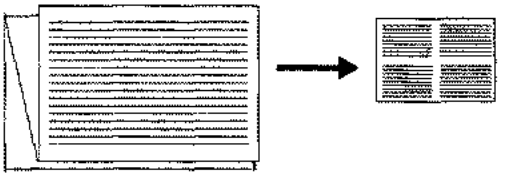
Listing Formats

By default, all RoadRunner operations produce a listing showing the names of files excluded from the operation for any reason. To customize the content of the listing and to produce multiple listings, use the **report** keyword. For each instance of **report**, a separate listing is produced, each with separate specifications: which fields are included, which

devices it is sent to, and a variety of page layout options. The **report** keyword is covered on page 5-55; this section gives some examples of RoadRunner's reporting capabilities.

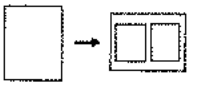
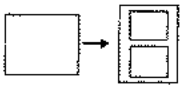
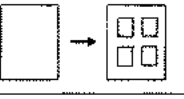
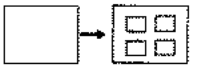
Format Options

Line printer listings are usually printed on 11" x 17" paper. For a very long listing, this can be wasteful in terms of resources and storage space. If you have a Hewlett-Packard LaserJet or compatible printer, RoadRunner can print the listings in a reduced typeface, grouped two or four to a sheet of 8 1/2" x 11" paper. You can use either portrait or landscape orientation and can print on the front and back of the sheet if your printer has duplex capability.



When archiving listings, it is useful to print two or four reduced page images per sheet.

The following chart shows how some of the report options work:

Report option	Layout
`report (format 2up,portrait)` 2 portrait-oriented images per sheet	
`report (format 2up,landscape)` 2 landscape-oriented images per sheet	
`report (format 4up,portrait)` 4 portrait-oriented images per sheet	
`report (format 4up,landscape)` 4 landscape-oriented images per sheet	

Use the **landscape** option to print 132-column data (for example, if you have used the **long field** option).

To print images on both sides of the paper, add the **duplex** option. The example following prints four pages on each side:

```
report (format 4up,landscape,duplex)
```

To use a custom environment file, specify the `environment` option as shown in the following example.

```
report (format environment graybar.envs.dp)
```

HP 2680A and Unison Formation users can do reduction printing using the standard HP environment files. When using an environment file, you cannot specify the **2up**, **4up**, **duplex**, **portrait** or **landscape** options. You can use **width** and **lines**.

```
report (to hp2680a format environment lp2.hpenv.sys)
```

Examples

The following job sends a default listing to a LaserJet (device #25) in 2-up format. A second listing is sent to a file called backlist, including:

- The command used to create the backup
- The volume IDs of the ANSI labeled tapes written
- The contents of the backup header backup date and time and other key information; for an example, see "Backup Header," on page 5-58.

```
!job jfullbak,operator.sys
!file ljet;dev=25
!file backlist,old;acc=append
!run rr.pub.tym
full to 7 dbstore
report (format 2up, duplex to *ljet)
report (with command, volnames, header to *backlist)
/go
exit
!eoj
```

The first use of `report` in the following example creates a default listing to `SYSLIST`. The second use writes the names of any files not restored to a disc file for review.

```
!job jglbak,operator.sys
!file rejects;dev=disc
!run rr.pub.tym
select @.@.GL
restore from 7
report
report (with nofiles to *rejects)
/go
exit
!eoj
```

If you are running RoadRunner from a terminal, the default listing device is your terminal screen. You can use the **offline** option to send the listing to the system printer, as shown below. Also, a list of the filenames written and the ANSI labeled tapes used is sent to a disc file. If you need to restore a file from the backup set, you could look it up in this file. For each filename listed, the ID of the tape volume on which it starts is shown.

```
:run rr.pub.tym
mpe "file filelist;dev=disc"
full to 7
report (to offline)
report (fullname,volname to *filelist)
/go
```

Tips, Tweaks and Tuning

This section provides a variety of suggestions for advanced users of RoadRunner. These ideas can help you improve performance and take advantage of RoadRunner's powerful backup management and system management capabilities.

Archiving and Purging Unneeded Files

RoadRunner's **purge** keyword removes selected files from your system after they are stored. This provides a way to do disc-space housekeeping on a regular basis. For example, to get rid of all EDITOR and QEDIT work files after each full backup, you might do the following:

```
select (K######.@.@,Q######.@.@ purge)
select (@.@.@ excluding K######.@.@,Q######.@.@)
full to tape
```

In the following example, files in the PUB.SYS account that were created since the last full backup by a user other than MANAGER.SYS are purged. After each file is purged, its name is written to a file. The RoadRunner system-defined parameter **\filename** captures the name of the file currently being processed. This example also illustrates parallel backup to ANSI labeled tapes.

```
mpe "file purgelst,old;acc=append"
select (@.pub.sys where credate > lastfull and creator <> manager
purge after each "echo \filename >> *purgelst")
select (@.pub.sys)
select (@.@.@ excluding @.pub.sys)
full to (tape label ansi name f1) and (tape label ansi name f2)
```

The next example shows how you might purge files that haven't been accessed in over a year:

```
select (@.@.@ where accdate < today-365 purge)
select (@.@.@ where accdate >= today-365)
store to tape
```

Controlling the Order in which Files are Stored

Because RoadRunner stores files in the order in which they are specified in the command, you can put accounts from which files are more frequently restored at the start of the tape set by placing those file sets first:

```
select (@.@.acctg,@.@.maillist)
select (@.@.@ excluding @.@.acctg,@.@.maillist)
partial to dat
```

The acctg and maillist accounts will be at the front of the tape.

Application Checkpointing

When you are about to make changes to a production account, it is often a good idea to make a copy of the affected files in case an error occurs during processing. This is called application checkpointing. If you have enough space to create the backup on disc, you can take advantage of RoadRunner's virtual tape backup capability to expedite this process. In the following example, the JANUARY.ACCTG group is stored to a disc file:

```
select @.january.acctg
store to (disc name vtjan93)
```

If you need to restore the backup, do the following:

```
select @.january.acctg
restore from (disc name vtjan93)
```

If the changes are completed successfully, delete the virtual tape file:

```
run rr.pub.tym
vtpurge vtjan93
```

Configuring Compression:Performance Tuning

Pro Module

Because compression is a CPU-intensive task that yields greater benefits with some data files than others and because the importance of reducing media usage varies from site to site, RoadRunner lets you select regular compression (approximately 2:1), high compression (approximately 4:1; Pro Module only), or no compression. This gives you control over the balance between media usage and elapsed time, and helps to facilitate unattended backup. For syntax, see the **compress** keyword on page 5-17.

A lower compression level increases the amount of tape needed to hold the compressed data, but decreases the CPU time the process requires. In other words, selecting a low level of compression could make the backup run faster, depending on the speed of the backup target device relative to the power of the CPU.

A higher compression level decreases the amount of media needed to hold the compressed data, but could increase the amount of time required for the backup if the CPU is not able to keep up with the target device. On very powerful processors, this is not a problem. Choosing a high level of compression can help implement tape- or disk-based unattended backup and can also reduce tape usage with traditional backup.

Following are some reasons why you might want to use increased compression:

- Unattended backup. Shops with DAT or 8-mm tape drives may find that the highest levels of compression allow them to store all their data on a single tape, eliminating the need for an operator to change tapes.
- Streamlining remote backup. Shops that back up remote machines and then send the virtual tape files over a network find the highest compression levels reduce the amount of data that must be transmitted over the network, saving time and communication costs.
- Additional time savings. If the CPU is very fast compared to the backup device (for example, a Series 980 with a 7978 tape drive, a 960 with a 7979 tape drive, or many CPUs with DDS drives), additional time savings are likely.
- Reducing tape use. Shops with powerful CPUs who want to reduce the number of tapes used for backup find that the highest compression levels usually cuts the number of tapes by a factor of four.
- Appended backup sets to DAT. With a storage capacity of over seven million sectors, high-capacity DAT drives allow you to use RoadRunner's compress and append capabilities to put a whole set of full and incremental backups on one tape.

RoadRunner allows compression levels to be set differently for selected filesets. This allows you to specify high compression for files that yield the most effective compression (like database files) and low or no compression for those that are poor candidates (like program files or QEDIT files). This way you do not waste CPU time on trying to compress files that have no unused space.

The following select criteria specifies maximum compression for IMAGE database files and standard compression for all other files during a full backup.

```
select (@.@.@ where isimage compress 2)
select (@.@.@ where <> (not) is image compress 1)
```

- The **report** (long, compression) option provides you with information on the compression level specified, the percentage compression achieved, and throughput in megabytes per second achieved for each file. Over time you may be able to target certain filesets where high compression is not effective. You can then try storing just those filesets with **compress=1**. If compression results are still poor, try **compress=0**.

Checking Whether you have Enough Space for a Restore

If you run out of disc space while restoring files, RoadRunner prints a message telling you how much more disc space you need to finish the restore and then terminates operation. If you know you have limited free space, you can prevent this situation from occurring by finding out if you have enough space before beginning the restore. Do the following:

```
select @.jan.gl listdir from tape report (with nofiles)
```

You then check the RoadRunner CI variable **rrsectorsexpected** and compare it to free disc space using the MPE DISCFREE utility.

Improving Disc Space Usage on Restore

Due to the way file space is allocated, many MPE/iX files are built using far more space than necessary for the data in the file. To return that space to the system without changing the file's potential maximum size, use the **trim** keyword when restoring files. To do this, the **trim** keyword invokes an option of MPE/iX's :FCLOSE intrinsic. Use it as follows:

```
select @.@.myacct trim restore from tape
```

You can also use it on a reload:

```
reload trim
```

MPE/iX Variable Substitution

To utilize MPE/iX variable substitution capabilities at any point in a RoadRunner command, include the variable in the RoadRunner command. This is most useful, of course, if RoadRunner is run from a job stream, so that the current variable value is factored in automatically at run-time.

Setting Execution Priority

In the following command, the job priority for the RoadRunner backup is set at cs, if the system time is after 6 p.m. If the system time is before 6, the job priority is set at ds.

```
!job backup, manager.sys
! if hphour > 18
!setvar pri "cs"
!else
!setvar pri "ds"
!endif
!run rr.pub.tym
full to tape pri !pri
/go
exit
!eoj
```

Day-Sensitive Backup

You can also create jobstreams that do full or partial backups according to what day it is:

```
!job backup,manager.sys
!if hpday = 6
!setvar command "full"
!else
!setvar command "partial"
!endif
!run rr.pub.tym
!command to tape
/go
exit
!eoj
```

Configuring Compression Based on Disc Space Availability

The following example uses the **scan** command to estimate the number of sectors required for unattended backup. If the space required is such that low compression can be used to achieve unattended backup, low compression is used. If not, high compression is used. Use this function if you have a great deal of batch processing you want to complete after the backup. In this situation, your top priority is to get the backup done as quickly as possible. Since low compression uses less CPU time (if you have a fast tape drive, not a DAT drive), it is the most time-conserving approach.

```
!job backup,manager.sys
!run rr.pub.tym
select @.@.@ where moddate >= lastfull scan
/go
exit
!if !rrsectorsexpected > 100000
!setvar compress = 2
!else
!setvar compress = 1
!endif
!run rr.pub.tym
partial to tape compress !compress
/go
exit
!eoj
```

Store and Validate in One Step

If you know your backup requires only one media volume or the number of volumes equal to the number of drives in use, you can store and validate in one step.

```
!job backup,manager.sys
!run rr.pub.tym
partial to 17 and 19
/go
validate from 17 and 19 autoload 120 autoreply
/go
exit
!eoj
```

Creating Test Accounts

You can duplicate production accounts to use in a testing environment. RoadRunner can simplify the creation of test accounts with the **directory** and create **usedir** options. In this example, the production accounts are stored to and restored from virtual tape files on disc. When the operation is complete the virtual tape files are purged.

```
!job backup,manager.sys
!run rr.pub.tym
select @.@.gl where isimage
store to (disc name vtdbase) directory minimum
/go
mpe "purgeacct testgl"
restore from (disc name vtdbase) change acct gl to testgl
create usedir
/go
vtpurge vtdbase
/go
exit
!eoj
```

See "Change Creator Modifications for MPE 5.0 and Later" on page 5-10. for specific information for creating test accounts on systems running MPE/iX 5.0 and later.

Error Recovery

Store Errors

RoadRunner's error recovery feature can continue a store after a write error or power failure without your having to abort and restart the operation. To control tape quality, you can specify a soft error threshold with the **maxerrs** keyword, after which RoadRunner rejects the current tape and invokes the recovery procedures used with unrecoverable errors.

Progress Reporting

If an irrecoverable error occurs on a particular volume, requiring that reel or cartridge be discarded and replaced with another, the messages reporting sector count and percent completion of the operation will be inaccurate. This is because some files have been written twice, once to the discarded reel and once to the replacement reel. Due to the nature of progress reporting, RoadRunner is unable to reset the progress percentage; completion is eventually shown as greater than 100%. This is normal in the situation described.

Restore Errors

There are three areas on a RoadRunner tape: the header that identifies the tape as a RoadRunner tape; the directory of files; and the data. Media errors can occur in any of these areas; RoadRunner provides recovery options for all of them. Also, RoadRunner provides recovery if you are missing reels from a tape set.

Directory Problems

You typically begin a restore by mounting the last volume of a multi-volume tape set. If a media error occurs during RoadRunner's attempt to read the directory of this tape, the following message is returned:

```
A media error occurred while reading the directory on ldev n
Either start the operation with a different volume or use
RECOVER MEDIA
```

At this point, you can restart the restore with the **recover media** option. The procedure to follow depends on whether you are dealing with a a single-volume set or multi-volume tape set, where there are other volumes which may contain a valid directory.

Always use the **concurrent** keyword when using the **recover media** option. The files restored will be printed as they are restored.

If another volume is available, mount it and restart the restore with the **recover media** option. For example:

```
restore from 7 select @.@.dallas recover media concurrent
```

With recover media, the index of files stored on the front of the store is not used. You must read the tape to the end and will be prompted with the following:

```
"Continue restore operation on 1 device?"
```

If all the files requested have been restored or this is the last media in the file set, reply "no."

The list of files restored will be in the STDLIST if you have used the **concurrent** keyword as suggested.

Tape Not Recognized as a RoadRunner Tape

Once the directory has been restored and the retrieval of files is in progress, mounting a volume with a corrupted header produces this message:

```
Media on ldev n not recognized as RoadRunner backup
```

If you receive this message and the incorrect reel was mounted, mount the correct reel. However, if the correct tape is not recognized by RoadRunner, you should `:ABORTIO` to the tape drive and restart the restore with **recover media**. When you get to the problem reel, RoadRunner displays the following message:

```
Media on ldev n not recognized as RoadRunner backup
Continue operation on ldev n?
```

If you choose to continue, RoadRunner scans the front of the tape until it either finds a valid data block from the same backup (backup ID is the same) or until it gives up (encounters 20 end of file marks, 5 errors, or reads 2000 blocks without seeing a directory block).

Missing Volumes

Ordinarily, if any volume of a multi-volume tape set is lost, RoadRunner will not restore volumes out of sequence. To restore from the volumes you have, run your restore with **recover media**. When you mount a non-consecutive volume, the following messages are produced:

```
Incorrect media volume (volume n of set n) mounted on ldev n
Continue operation on ldev n?
```

Respond affirmatively to restore all possible files.

If you have only one volume, an alternate approach is to restore with the **single** keyword. This restores all available files and terminates after that reel.

```
restore from tape select @.@.@ single
```

Media Error Within Files

When a media error occurs within a file, the following message is produced:

```
A backup media error occurred while restoring file filename
```

Restart the restore with the **recover files** option to recover as much data as possible. For instance, if there is a 500,000 sector dataset that cannot be restored because one block is corrupt, **recover files** allows as much of the dataset to be restored as possible and prints a warning to alert you that a corrupt file may be introduced to your system.

Combined Recovery Procedures

recover all can be used to invoke both types of recovery. One situation where this could be useful is if part of a tape has been overwritten, and you need to restore anything possible from it. In this case, **recover all** tells RoadRunner to accept the reel, find where the data begins, and restore the files even if they are incomplete.

Chapter

4 Commands

This section provides complete details on all RoadRunner commands.

Overview

A RoadRunner command is composed of a command and various optional keywords that modify that command. RoadRunner's commands are as follows:

Command	Description
`copy`	Duplicates all or part of a previous backup (page 4-2).
`full`	Stores all user and system files (page 4-4).
`fulldate`	Checks or changes the full backup date (page 4-5).
`incremental`	Stores files modified since last interim backup (page 4-6).
`interim`	Stores files modified since last partial backup (page 4-7).
`interimdate`	Checks or changes the interim backup date (page 4-8).
`listdir`	Displays file directory of a backup (page 4-9).
`listinfo`	Lists the command used to initiate a backup, the backup header, and the volume IDs of any labeled tapes used (page 4-11).
`partdate`	Checks or changes the partial backup date (page 4-12).
`partial`	Stores files modified since full backup (page 4-13).
`reload`	Restores all files, directories, UDCs (page 4-14).
`restore`	Retrieves files from RoadRunner backups (page 4-15).
`scan`	Lists files selected by selection criteria (page 4-18).
`store`	Copies files to tape and virtual tape filesets (page 4-19).
`validate`	Verifies accuracy and restorability of tape set (page 4-22).
`vtpurge`	Deletes virtual tape filesets (page 4-24).
`vtrename`	Changes the name of virtual tape filesets (page 4-25).

Some of the keywords used with these commands specify basic information like the target fileset (**select**) or execution priority (**priority**). Other keywords control sophisticated RoadRunner features like the event/action mechanism (**after**) or formatting options for

listings (**report**). The following example shows a RoadRunner command, consisting of a **store** command with leading and trailing keywords:

```
select @.acct.dev, r@.@.jan
select (@.@.myacct where moddate=today-4)
store to (tape name backup) buffer 50% report (format 2up)
```

The command and its keywords can be specified in any order. Most keywords can be repeated.

copy

Copies all or part of any existing RoadRunner backup. The selected files are duplicated in RoadRunner format. Use **copy** to duplicate backups from tape to tape, tape to disc, disc to tape, tape to DAT, DAT to tape, and DAT to DAT. It can also be used to archive selected portions of a backup; for example, only user files and not applications, or only certain accounts.

If you enter **copy** without a **select** statement, the default fileset is all files and does not depend on your MPE/iX access capabilities.

If you do not specify **from** or **to** devices, RoadRunner issues a request to the system console for the file RRSRC and RRDEST, respectively.

If the source backup includes the system directory, it is included in the copy. To exclude it, specify **nodirectory**.

When copying an appended backup from a DAT cartridge, you must specify the index number of the backup to be copied with the **index** keyword. To append a copied backup to a DAT cartridge that already contains backups, use the **append** keyword to place it at the end of the data already there.

Examples

The following command copies files in a backup from logical device 17 to logical device 7, with requests for tape files named "source" and "dest", respectively.

```
copy from (17 name source) to (7 name dest)
```

The following command copies only the gl account from a previous backup to a new backup. Since only the **device** class is specified on the **from** and **to** keywords, requests for specific logical devices are sent to the system console. The command also purges any existing file named outfile. After the operation is complete, it sends a list of copied files in the short format and written to outfile.

```
select @.@.gl
copy from tape to tape
mpe "purge outfile" mpe "file outfile; rec=-80"
report (short to *outfile)
```

The following command copies the fourth backup from a DAT backup set and appends it to the end of another DAT backup set.

```
copy from dat index 4 to dat append
```

The following command copies the RoadRunner backup from a tape on logical device 7 to a tape on logical device 8 without requiring a console reply.

```
copy from 7 to 8 autoreply
```

The following command copies the virtual tape backup to a tape on logical device 7.

```
copy from (disc name RRVT.G.A.) to 7 autoreply
```

Keywords Keywords used with **copy** are listed below.

Keyword	Description
`after`	Designates an MPE/iX command to be performed when a specified event occurs (page 5-1).
`append`	Stores an additional backup to a DAT cartridge (page 5-4).
`autoeject`	Ejects DAT cartridge automatically after operation (page 5-6).
`autoload`	Brings the specified logical device on-line (page 5-6).
`autoreply`	Configures tape drive to auto-reply (page 5-8).
`backupname`	Specifies name of the backup being written (page 5-9).
`combine`	Combines source sets created in parallel (page 5-16).
`concurrent`	Prints reports as files are processed (page 5-18).
`description`	Specifies text written to the backup header (page 5-27).
`nodirectory`	Excludes MPE/iX directory if present (page 5-28).
`display`	Controls program messages (page 5-31).
`excluding`	Excludes files from a fileset (page 5-33).
`from`	Specifies the device on which the source fileset resides (page 5-35).
`include`	Incorporates a text file into the command (page 5-38).
`mpe`	Executes MPE/iX commands (page 5-45).
`maxerrs`	Specifies the number of soft errors allowed before RoadRunner initiates recovery (page 5-44).
`nopriv`	Sets the file code which RoadRunner uses for creating virtual tape backups to a positive (nonprivileged) value (page 5-46).
`priority`	Specifies an MPE/iX execution priority (page 5-51).
`report`	Controls listing (page 5-55).
`noroadrest`	Omits RoadRunner restore module (page 5-47).
`select`	Specifies the source filesets (page 5-60).
`to`	Specifies one or more destination devices (page 5-69).
`where`	Qualifies source filesets by file attributes (page 5-78).

full

Backs up the entire system, including the MPE/iX directory or directories including account capabilities and UDCs. It resets the full, partial, and interim backup dates and times in the file bpfbdate.pub.sys to the date and time at the beginning of the backup.

If you enter **full** without a **select** statement, the default fileset depends on your MPE/iX access capabilities. If you have SM or OP capability, you can store all files (@.@.@ or /); if you have AM, you can store only your account; otherwise, you can store only your logon group.

If you do not specify a to device, RoadRunner generates a request to the system console for a logical device in class TAPE.

Although this command stores IMAGE database files, the dirty bit and backup date in the root file are not reset unless you specify **dbstore**.

WARNING

If you use the **select** keyword with this command, it overrides the default file selection. If you intend to store all the files on your system, make sure the sum total of your selects adds up to all files on your system.

For example:

```
full to tape
select (@.pub.sys after all "tellop pub.sys backup complete")
select (@.@.accting, @.@.mfg dbstore)
select (@.@.@ excluding @.pub.sys, @.@.accting,@.@.mfg)
```

The first two selects establish special processing for certain accounts. The third select is required to include other accounts in the backup.

Examples

The following command backs up all files and directories on the system. It generates a request on the console for a logical device in class TAPE.

```
full
```

The following command backs up everything on the system except native mode program files, to a tape file named backup. RoadRunner generates a request for the logical device where the backup is mounted at the system console.

```
full to (tape name backup)
excluding (@.@.@ where code = nmprog)
```

Keywords

See the keyword list for the **store** command on page 4-19.

fulldate

Displays and changes the date and time of the last full backup stored in the file bpfbdate.pub.sys, or in an alternate file specified with the **datefile** keyword.

```
fulldate [mm/dd/yy[at hh:mm]] [datefile filename]
```

If you do not specify a date, **fulldate** displays the date and time of the last full backup in native language support (NLS) format. If you specify a date and time, **fulldate** changes the date and time to the one specified. If you do not include a time, the default is 12:00 midnight.

Time is entered and displayed in 24-hour format.

For information on the **datefile** keyword, see page 5-22.

Examples

The following command uses **fulldate** without a date. RoadRunner prints the date and time of the last backup.

```
:run rr.pub.tym
<01>fulldate
<02>/go
The last full backup was performed on 3/16/96 at 22:05
```

The next command uses **fulldate** with a date, but no time. RoadRunner changes the full backup date to the one specified and sets the time to 12 midnight.

```
:run rr.pub.tym
<01>fulldate 4/4/96
<02>/go
The last full backup was performed on 04/07/96 at 00:12
The full backup date has been set to 04/04/96 at 00:00
```

Use the **fulldate** command only if the file (bpfbdate) has been created from running RoadRunner using the **full** keyword.

incremental

Backs up: 1) files modified since the last interim backup, and 2) system directories. To do this, it checks the interim backup date and time in bpfbdate.pub.sys. If no **interim** backup has been performed since the last **partial** or **full**, all files modified since are stored. After the operation is complete, it resets the interim backup date and time to the date and time at the beginning of the backup. For more information on the different backup types, see page 2-4.

If you enter **incremental** without a **select** statement, the default fileset depends on your MPE/iX access capabilities in addition to the date qualification. If you have SM or OP capability, you can store all files (@.@.@ or /); if you have AM, you can store only your account; otherwise, you can store only your logon group.

WARNING

If you use the **select** keyword with this command, it overrides the default file selection. If you intend to store all the files on your system, make sure the sum total of your selects adds up to all files on your system.

For example:

```
incremental to tape
select (@.pub.sys after all "tellop pub.sys backup complete")
select (@.@.accting, @.@.mfg dbstore)
select (@.@.@ excluding @.pub.sys, @.@.accting,@.@.mfg)
```

The first two selects establish special processing for certain accounts. The third select is required to include all other accounts in the backup.

If you do not specify a **to** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

Although this command stores IMAGE database files, the dirty bit and backup date in the root file are not reset unless you specify **dbstore**.

Examples

The following command backs up files modified since the last interim backup and system directories. It generates a request on the console for a logical device in class TAPE.

```
incremental
```

The next command backs up date-qualified files on the system except for native mode program files; it stores them to a logical device named backup in device class TAPE.

```
incremental to (tape name backup)
excluding (@.@.@ where code = nmprog)
```

Keywords

See the keyword list for the **store** command on page 4-19.

interim

Backs up: 1) files modified since the last partial backup, and 2) system directories. To do this, it checks the partial backup date and time in bpfbdate.pub.sys. After the operation is complete, it resets the interim backup date and time to the date and time at the beginning of the backup.

If you enter **interim** without a **select** statement, the default fileset depends on your MPE/iX access capabilities in addition to the date qualification. If you have SM or OP capability, you can store all files (@.@.@ or /); if you have AM, you can store only your account; otherwise, you can store only your logon group.

WARNING

If you use the **select** keyword with this command, it overrides the default file selection. If you intend to store all the files on your system, make sure the sum total of your selects adds up to all files on your system.

For example:

```
interim to tape
select (@.pub.sys after all "tellop pub.sys backup complete")
select (@.@.accting, @.@.mfg dbstore)
select (@.@.@ excluding @.pub.sys, @.@.accting,@.@.mfg)
```

The first two selects establish special processing for certain accounts. The third select is required to include all other accounts in the backup.

If you do not specify a **to** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

Although this command stores IMAGE database files, the dirty bit and backup date in the root file are not reset unless you specify **dbstore**.

Examples

The following command backs up files modified since the last partial backup and system directories. It generates a request on the console for a logical device in class TAPE.

```
interim
```

The following command backs up all date-qualified files on the system except for native mode program files; it stores them to a logical device named backup in device class TAPE.

```
interim to (tape name backup)
excluding (@.@.@ where code = nmprg)
```

Keywords

See the keyword list for the **store** command on page 4-19.

interimdate

Displays and changes the date and time of the last interim backup stored in the file bpfb-date.pub.sys, or in an alternate file specified with the **datefile** keyword.

interimdate [*mm/dd/yy*[**at** *hh:mm*]] [**datefile** *filename*]

If you do not specify a date, **interimdate** displays the date and time of the last interim backup in native language support (NLS) format. If you specify a date and time, **interimdate** changes the date and time to the one specified. If you do not include a time, the default is 12 midnight.

Time is entered and displayed in 24-hour format.

For information on the **datefile** keyword, see page 5-22.

Examples

The following command uses **interimdate** with no date. RoadRunner displays the date and time of the last interim backup.

```
:run rr.pub.tym
<01>interimdate
<02>/go
The last interim backup was performed on 01/16/96 at 22:05
```

The following command uses **interimdate** with a date, but no time. RoadRunner changes the interim backup date to the one specified, and sets the time to 12 midnight.

```
:run rr.pub.tym
<01>interimdate 4/4/96
<02>/go
The last interim backup was performed on 04/07/96 at 00:12
The interim backup date has been set to 04/04/96 at 00:00
```

listdir

Produces a listing of the RoadRunner file directory for a backup tape set or virtual tape fileset.

To obtain a listing of all the files on a multi-volume tape set, mount the last reel of that set. The directory on this tape contains information for all files stored on all reels of the tape set. Previous tapes contain the cumulative directory as it stood when that particular tape was completed. (The reel number for files stored on the current and subsequent reels is recorded as "0" in the directory of those intermediate reels).

To obtain a listing of files in a virtual tape backup, use the **from** keyword to specify the root file. Use **select** to describe a source fileset if you do not want to list all the files in the backup.

If you do not specify the **report** keyword, **listdir's** default output is a listing of the files under the short format. See page 5-22 for details on the **report** keyword.

If you enter **listdir** with no keywords, a request for a logical device in device class TAPE is sent to the system console.

Examples

The following command lists the directory for a backup on tape. Since no tape drive is specified, a request for a logical device in device class TAPE is sent to the system console.

```
listdir
```

The following command lists the directory for the third backup appended to a DAT cartridge.

```
listdir from 7 index 3 report (fullname with header)
```

The following command lists the backup directory from a virtual tape file. The root file is back-referenced with a file equation.

```
:file xxvt;dev=disc
listdir from *xxvt
```

The next command indicates the root file using the name parameter of the **from** keyword. Only the files in mygroup.myacct are listed.

```
select @.mygroup.myacct
listdir from (disc name xxvt)
```

The following command reads the backup directory on logical device 7 and looks for IMAGE files in the prod account with more than 50,000 sectors. A report listing those files, and showing the specified fields for each, is sent to the disc file dblist.

```
select (@.@.prod where is image and space >50000)
listdir from 7
mpe "file dblist; rec= -80,f,ascii"
report (fullname, sectors, code, compression with command,
header to *dblist)
```

A sample **listdir** command and the default listing produced is shown below.

```
listdir from (disc name vt)
/go
Starting Directory Build
Finished Directory Build
6 files on this backup
```

```
  3 directories on this backup
  11 files selected to be listed

FILENAME.GROUP.ACCOUNT  VOLUME RESTRICTIONS SECTORS CODE  MEDIA
AFTER    .SCOTT   .SALES    DISC               :C      16     0
AFTER2   .SCOTT   .SALES    DISC               :C      16     0
ALLBACK  .SCOTT   .SALES    DISC               :C      16     0
BLESSED  .SCOTT   .SALES    DISC               :C      16     0
BPFBDATE.SCOTT    .SALES    DISC               :C      16     0
CREST    .SCOTT   .SALES    DISC               :C      16     0

6 files were listed
```

Keywords Keywords used with **listdir** are listed in the following table.

Keyword	Description
autoeject	Ejects DAT cartridge automatically after operation (page 5-6).
autoload	Brings the specified logical device on-line (page 5-6).
autoreply	Configures tape drive to auto-reply (page 5-8).
display	Controls program messages (page 5-31).
excluding	Excludes files from a fileset (page 5-33).
from	Specifies the device on which the source fileset resides (page 5-35).
include	Incorporates the content of a text file into the command (page 5-38).
index	Specifies which backup of an appended backup set to use (page 5-39).
mpe	Executes MPE/iX commands (page 5-45).
priority	Specifies an MPE/iX execution priority (page 5-51).
report	Specifies format, content and destination of the listing (page 5-55).
select	Specifies the source fileset (page 5-60).
where	Qualifies source filesets by file attributes (page 5-78).

listinfo

Reads a backup tape or virtual tape fileset and produces a listing of the command used to initiate a backup, the backup header, and the volume IDs of any labeled tapes used. It is equivalent to specifying the **listdir** command with the following options:

```
listdir
report (with header, command, volnames, nofiles, norejects)
```

The backup header listed by listinfo contains basic information about the current backup. See "Backup Header" on page 5-58 for an example. Among other things, it shows the volume number and media set number (relevant only when multiple output paths are used in parallel). For appended backups, it shows which backup this is and what volume of the backup set it starts on.

```
This is volume a of media set b
Backup number is x, first volume of backup is y
```

The content of the **listinfo** report is pre-defined, so you cannot specify fields or **with** options. The **report** keyword is used to specify only the destination(s) and format of the report.

If you enter **listinfo** with no keywords, a request for a logical device in device class TAPE is sent to the system console.

Examples

The following command lists the command, header and volume IDs for a backup on tape. Since no tape drive is specified, a request for a logical device in device class TAPE is sent to the system console.

```
listinfo
```

The following command specifies a particular printer for the **listinfo** report.

```
listinfo from 7
report (to 25)
```

Keywords

Keywords used with **listinfo** are listed in the following table.

Keyword	Description
`autoeject`	Ejects DAT cartridge automatically after operation (page 5-6).
`autoload`	Brings the specified logical device on-line (page 5-6).
`autoreply`	Configures tape drive to auto-reply (page 5-8).
`from`	Specifies the device on which the source fileset resides (page 5-35).
`index`	Specifies which backup of an appended backup set to use (page 5-39)
`report`	Specifies format and destination of the listing (page 5-55).

partdate

Displays and changes the date and time of the last partial backup stored in the file bpfb-date.pub.sys, or in an alternate file specified with the **datefile** keyword.

```
partdate [mm/dd/yy[at hh:mm]] [datefile filename]
```

If you do not specify a date, **partdate** displays the date and time of the last partial backup in native language support (NLS) format. If you specify a date and time, **partdate** changes the date and time to the one you specify. If you do not include a time, the default is 12 midnight.

Time is entered and displayed in 24-hour format.

For information on the **datefile** keyword, see page 5-22.

Examples This command uses **partdate** without a date. RoadRunner displays the date and time of the last backup.

```
run rr.pub.sys
<01>partdate
<02>/go
The last partial backup was performed on 03/16/96 a 22:05
```

The following command uses **partdate** with a date, but no time. RoadRunner changes the partial backup date to the one specified, and sets the time to 12 midnight.

```
run rr.pub.sys
<01>partdate 4/4/96
<02>/go
The last partial backup was performed on 04/07/96 at 00:12
The partial backup date has been set to 04/04/96 at 00:00
```

partial

Backs up 1) files modified since the last full backup, and 2) system directories. To do this, it checks the full backup date and time in the file bpfbdate.pub.sys. After the operation is complete, it resets the partial and interim backup dates and times to the current system date and time.

If you enter **partial** without a **select** statement, the default fileset depends on your MPE/iX access capabilities in addition to the date qualification. If you have SM or OP capability, you can store all files (@.@.@ or /); if you have AM, you can store only your account; otherwise, you can store only your logon group.

WARNING

If you use the **select** keyword with this command, it overrides the default file selection. If you intend to store all the files on your system, make sure the sum total of your selects adds up to all files on your system.

For example:

```
partial to tape
select (@.pub.sys after all "tellop pub.sys backup complete")
select (@.@.accting, @.@.mfg dbstore)
select (@.@.@ excluding @.pub.sys, @.@.accting,@.@.mfg)
```

The first two selects establish special processing for certain accounts. The third select is required to include all other accounts in the backup.

If you do not specify a **to** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

Although this command stores IMAGE database files, the dirty bit and backup date in the root file are not reset unless you specify **dbstore**.

Examples

The following command generates a backup of all files modified since the last full backup and the system directories. It generates a request on the system console for a logical device in device class TAPE.

```
partial
```

The following command backs up all date-qualified files on the system except for native mode program files, and stores them to a tape device named backup.

```
partial to (tape name backup)
excluding (@.@.@ where code = nmprog)
```

Keywords

See the keyword list for the **store** command on page 4-19.

reload

Restores all files, and MPE/iX directories from a RoadRunner backup made using either **full, partial, incremental, interim** commands, or the **directory** keyword. See "Database Backup and Restore," on page 2-7.

When you enter **reload** with no keywords, RoadRunner rebuilds the directory structure, restores user defined commands (UDCs), and restores all files for which you have access capabilities from the backup. The backup must have been made using either **full, partial, incremental, interim** commands, or the **directory** keyword. If you have SM or OP capability, you can restore all files (@.@.@ or /); if you have AM, you can restore only your account; otherwise, you can restore only your logon group. Use of the **select** keyword overrides the default fileset and only the files specified in the **select** statement(s) are restored.

It maintains the access and modification dates of the restored files and does not overwrite a file already existing on disc with the same name. **reload** is equivalent to entering:

```
select @.@.@
restore from tape olddate keep directory all
```

nokeep, nodirectory and **newdate** can be used to override these defaults.

If you do not specify a **from** device, **reload** sends a request for a logical device in class TAPE to the system console.

Note

Whenever you perform a store with the **directory** keyword, make sure you also store COMMAND.PUB.SYS. This file describes your system UDC structure. When you restore a particular version of your system directory using the **udcs** option, you *must* restore the corresponding copy of COMMAND.PUB.SYS to avoid UDC initialization errors. After the restore, log off and log on again to activate the restored UDCs.

Examples

Since no source device is specified, the following command generates a request to the system console for a logical device in class TAPE. Once the device is available, RoadRunner proceeds with the reload as described above.

```
reload
```

The following command reloads all files from the backup on logical device 7. The first **select** statement specifies a group of files that are to overwrite existing versions on disc.

```
select (command.pub.sys, @.net.sys, nmconfig.pub.sys nokeep)
select (@.@.@ excluding command.pub.sys, @.net.sys, nmcon-
fig.pub.sys)
reload from 7
```

Keywords

See the keyword list for the **restore** command on page 4-15.

restore

Retrieves files from a RoadRunner backup.

If you enter **restore** without a **select** statement, the default fileset depends on your MPE/iX access capabilities. If you have SM or OP capability, you restore all files (@.@.@ or /); if you have AM, you restore only your account; otherwise, you restore only your logon group.

Note

Parallel stores and restores are not supported over a network.

If you do not specify a **from** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

To ensure that the access and modification dates of the files being restored remain unchanged, always add the **olddate** keyword to your **restore** command. If you do not specify **olddate**, RoadRunner changes the dates of restored files to the date and time of the restore.

When restoring selected files from a multi-volume tape set, mount the last reel first. Because the directory on the last volume contains the complete directory for the backup set with the location of all files, this is the most efficient approach.

Example

The following command restores all files in the group april.gl from the backup on logical device 7:

```
select @.april.gl
restore from 7
olddate
/go
```

The following jobstream performs a restore:

```
!job jrestore,operator.sys
!run rr.pub.tym
   select @.pub.data
   restore from 7
   report
   display all
   olddate
   /go
   exit
!eoj
```

If you have done a parallel store, use parallel restore when performing a reload or restoring a large fileset from the backup. If you are restoring a small fileset or a single file, perform a single path restore. Parallel stores and restores are not supported over a network.

Keywords

Keywords used with **restore** and **reload** are listed in the following table. In the case of keyword pairs, the default is shown in bold, unless the default is different for **restore** and **reload**.

Keyword	Description
`after`	Designates an MPE/iX command to be performed when a specified event occurs (page 5-1).
`autoeject`	Ejects DAT cartridge automatically after operation (page 5-6).
`autoload`	Puts the specified logical device on-line (page 5-7).
`autoreply`	Configures tape drive to auto-reply (page 5-8).
`change`	Alters file, creator, group and/or account names (page 5-10).
`concurrent`	Prints reports as files are processed (page 5-18).
`copyacd` `nocopyacd`	Controls whether ACDs are restored (page 5-19).
`create`	Creates missing groups, accounts, and creators (page 5-20).
`dbrestore` `nodbrestore`	Controls special database processing (page 5-25).
`directory` `nodirectory`	Controls what aspects of the MPE/iX directory are included in the restore (page 5-28). **directory** is the default for **reload**; **nodirectory** is the default for **restore**.
`display`	Controls program messages (page 5-31).
`excluding`	Excludes files from a fileset (page 5-33).
`from`	Specifies the device on which the source fileset resides (page 5-35).
`include`	Incorporates a text file into the command at runtime (page 5-38).
`index`	Specifies which backup of an appended backup set to restore (page 5-39).

`keep`
`nokeep`
`keepnew`
	When restoring backup files with the same names as disc files, **keep** specifies that the files on disc are retained, **nokeep** specifies that the disc files are overwritten, and **keepnew** specifies that the newer of the two files is kept (page 5-41). **keep** is the default for **reload**; **nokeep** is the default for **restore**.

`local` `nolocal`	Controls whether files are restored into your group and account or their original groups and accounts (page 5-42).
`mpe`	Executes MPE/iX commands (page 5-45).
`newdate` `olddate`	Controls whether files' creation date is changed to the date of restore (page 5-48). **olddate** is the default for **reload**; **newdate** is the default for **restore**.
`priority`	Specifies an MPE/iX execution priority (page 5-51).
`purgeafter` `purgebefore`	Controls whether disc files are purged before or after identically named files have been restored (page 5-53).

Keyword	Description
recover	Restores files from a damaged tape or a tape set with missing reels (page 5-54).
report	Specifies format, content, and destination of the listing (page 5-55).
select	Specifies the source fileset (page 5-60).
single	Restores files from a single reel (page 5-65).
spoolpri	Specifies output priority for restored spool files. (page 5-68).
trim **notrim**	Controls resizing of file (page 5-71).
volclass	Restricts restored files to a specified volume class (page 5-74).
volset	Restricts restored files to a specified volume set (page 5-75).
volume	Restricts restored files to a specified volume (page 5-77).
where	Qualifies source filesets by file attributes (page 5-78).

scan

Evaluates selection criteria, reporting the number of files selected and the uncompressed sectors they comprise. Use **scan** to estimate the space needed for unattended backup without performing a backup or interrupting users. By default, no listing is produced.

Examples

When you plan to store databases, you can use the **dbstore** keyword with the **scan** command to ensure that the root files you've specified in your selects are properly expanded to include all data sets.

```
select acctdb,mandb scan dbstore report long
```

Keywords

The keywords used with **scan** are listed in the following table.

Keyword	Description
`dbstore`	Includes entire database if root file is specified (page 5-26).
`display`	Controls program messages (page 5-31).
`excluding`	Excludes files from a fileset (page 5-33).
`include`	Incorporates a text file into the command at runtime (page 5-38).
`onvs` `notonvs`	Controls whether files on specified volume sets are included (page 5-50).
`report`	Specifies format, content and destination of the listing (page 5-55).
`select`	Specifies the source fileset (page 5-60).
`splitvs`	Selects specified "backup files" for disk mirroring (page 5-66).
`where`	Qualifies source filesets by file attributes (page 5-78).

store

The **store** command backs up data to tape or virtual tape filesets. These backups are recoverable only by RoadRunner's **restore** and **reload** commands.

If you enter **store** without a **select** statement, the default fileset depends on your MPE/iX access capabilities. If you have SM or OP capability, you can store all files (@.@.@ or /); if you have AM, you can store only your account; otherwise, you can restore only your logon group.

If you do not specify a **to** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

Although this command stores IMAGE database files, the dirty bit and backup date in the root file are not reset unless you specify **dbstore**.

Spoolfiles that are printed while their storebit are on are not purged by MPE after printing. To ensure that these files are purged, exclude them from your **select** statement (page 5-60), or use the **nolock** keyword (page 5-43) on that local fileset to keep the storebits from being set. For example:

```
select (@.@.@ excluding @.out.hpspool)
select (@.out.hpspool nolock)
```

For other backup commands, see **full** (page 4-4), **incremental** (page 4-6), and **partial** (page 4-13), as well as "Routine Backups" on page 2-2.

Examples

The following command stores the selected files to a virtual backup file named check:

```
select @.version4.prod store to (disc name check)
```

The following command selects all files in the pub.sys group modified in the last four days and stores them to the media on logical device 7.

```
select @.pub.sys where moddate>today-4 store to 7
```

Keywords

Keywords used with **full, partial, interim, incremental** and **store** are listed in the following table. In the case of keyword pairs, the default is shown in bold, unless the default varies for the different commands.

Keyword	Description
after	Designates an MPE/iX command to be performed when a specified event occurs (page 5-1).
append	Stores an additional backup to a DAT cartridge (page 5-4).
autoeject	Ejects DAT cartridge automatically after operation (page 5-6).
autoload	Puts the specified logical device on-line (page 5-6).
autoreply	Configures tape drive to auto-reply (page 5-8).
backupname	Specifies the name of the backup being written (page 5-9).
change	Alters file, group, account or creator name (page 5-10).
compress	Sets the level of data compression (page 5-17).
concurrent	Prints reports as files are processed (page 5-18).

Keyword	Description
`copyacd` `nocopyacd`	Stores ACDs for selected files (page 5-19). Disables ACD /store for selected files (page 5-19).
`dbfast`	Eliminates check for data above high water mark in database backup (page 5-22).
`dbstore` `nodbstore`	Stores entire database if root file is specified (page 5-26). Disables **dbstore** commands (page 5-26).
`description`	Specifies text written to the backup header (page 5-27).
`datefile`	Specifies an alternate file in which to store and retrieve **full/partial/interim** backup dates, instead of RoadRunner's default file (page 5-22).
`directory` `nodirectory`	Controls whether a copy of the MPE/iX directory is included in the backup. **directory** is the default for **full, interim, partial**, and **incremental**; **nodirectory** is the default for **store** (page 5-28).
`display`	Controls program messages (page 5-31).
`excluding`	Excludes files from a fileset (page 5-33).
`include`	Incorporates a text file into the command at runtime (page 5-38).
`interleave`	Specifies the number of files to be read concurrently from disc (page 5-40).
`lock` `nolock`	Denies write access to selected files (page 5-43). Allows write access to selected files (page 5-43).
`mpe`	Executes MPE/iX commands (page 5-45).
`maxerrs`	Specifies the number of soft errors allowed before RoadRunner initiates recovery (page 5-44).
`nopriv`	Sets the file code which RoadRunner uses for creating virtual tape backups to a positive (nonprivileged) value (page 5-46).
`noroadreel`	Omits the copy of RoadRunner normally written at the start of the backup in MPE/iX STORE format (page 5-47).
`onvs` `notonvs`	Controls whether files on specified volume sets are included (page 5-50).
`priority`	Specifies an MPE/iX execution priority for backup (page 5-51).
`purge` `nopurge`	Disc files are purged after they have been stored (page 5-52). Disables purge command for selected files (page 5-52).
`report`	Specifies format, content, and destination of the listing (page 5-55).
`select`	Specifies the source fileset (page 5-60).
`splitvs`	Selects specified "backup files" for disk mirroring (page 5-66).

Keyword	Description
`to`	Specifies one or more destination devices (page 5-69).
`unlock`	Specifies when files are released for other processing (page 5-73).
`where`	Qualifies source filesets by file attributes (page 5-78).

validate

Verifies integrity and restorability of a RoadRunner-created tape set, checking the media block sequence and file directory for errors.

If you enter **validate** without a **select** statement, the default fileset is all files (`@.@.@` or `/`).

If you do not specify a **from** device, RoadRunner generates a request to the system console for a logical device in class TAPE.

If you do not specify the **report** keyword, the listing includes the names of any files that cannot be restored.

Examples

The following command validates the backup media on logical device 16. A list of non-restorable files is also sent to the file nogood. The **nofiles** option excludes the names of the restorable files from this listing.

```
mpe "file nogood;dev=disc"
validate from 16 report (with nofiles to *nogood)
```

The next command checks the restorability of IMAGE and KSAM XL files on the backup, and lists the names of restorable and non-restorable files to a LaserJet.

```
mpe "file ljet;dev=25"
select @.@.@ where isimage or isksamxl validate from 16 report
(to *ljet)
```

This command validates the third backup appended to a cartridge set:

```
validate from DAT index 3
```

Keywords

Keywords used with **validate** are listed in the following table.

Keyword	Description
`after`	Designates an MPE/iX command to be performed when a specified event occurs (page 5-1).
`autoeject`	Ejects DAT cartridge automatically after operation (page 5-6).
`autoload`	Brings the specified logical device on-line (page 5-6).
`autoreply`	Configures tape drive to auto-reply (page 5-8).
`concurrent`	Prints reports during validation (page 5-18).
`display`	Controls program messages (page 5-31).
`excluding`	Excludes files from a fileset (page 5-33).
`from`	Specifies the source device (page 5-35).
`include`	Incorporates a text file into the command at runtime (page 5-38).
`index`	Specifies which backup of an appended backup set to validate (page 5-39).
`mpe`	Executes MPE/iX commands (page 5-45).
`priority`	Specifies an MPE/iX execution priority for validation (page 5-51).
`recover`	Validates damaged tapes or tape sets missing reels (page 5-54).

Keyword	Description
report	Specifies format, content, and destination of listing (page 5-55).
select	Specifies the source fileset (page 5-60).
single	Validates files from a single reel (page 5-65).
where	Qualifies source filesets by file attributes (page 5-78).

vtpurge

Since the files used for virtual tape backup are privileged files, the MPE/iX PURGE command cannot delete them from disc. RoadRunner's **vtpurge** command deletes an entire virtual tape fileset when you specify the root file name.

```
vtpurge filename
```

In rare instances, you may want to purge only one file from a multiple file virtual tape backup. This is risky because purging individual backup files could cause a future restore operation to fail if any portion of a file to be restored was contained in the purged file. To purge an individual file, indicate the name of that file. You cannot purge only the root file.

When purging virtual tape files on a remote node, you must use a file equation to specify the files to purge.

Example

This command purges all files in the monday fileset, including monday, monday01, monday02, and so forth:

```
:run rr.pub.tym
<01>vtpurge monday
<02>/go
```

This command purges only monday02:

```
:run rr.pub.tym
<01>vtpurge monday02
<02>/go
```

The following command purges the monday fileset on a remote node called sys2:

```
:run rr.pub.tym
<01>mpe "file monday.pub.backup;dev=sys2#disc"
<02>vtpurge *monday
<03>/go
```

vtrename

Changes the name of a virtual tape fileset.

vtrename *fileset* **[to]** *fileset*

Since the files used for virtual tape backup are privileged files, the MPE/iX RENAME command cannot rename them. You can use RoadRunner's **vtrename** command to change a virtual tape fileset's file, group, and account. The new group and account must reside on the same volume set.

If you change the group or account name of files, the renamed files must all originate in the same group or account; otherwise, an error occurs. Lockwords are not allowed in the destination fileset name. If you do not specify an account or a group, the destination filesets retain their original groups and accounts. If you specify a group but no account, RoadRunner places the destination filesets in your logon account.

Example The following command renames the virtual tape fileset monday.backup to april.backup:

```
:run rr.pub.tym
<01>vtrename monday.backup to april.backup
<02>/go
```

or

```
:run rr.pub.tym
<01>vtrename monday.backup to april
<02>/go
```

To rename daily.backup.sys to your logon group and account:

```
:run rr.pub.tym
<01>vtrename daily.backup.sys daily
<02>/go
```

Chapter 5

Keywords

This section provides complete details on all RoadRunner keywords.

Overview

A RoadRunner command is composed of a command and various optional keywords that modify that command. RoadRunner's keywords are as follows:

Keywords

The keywords used with RoadRunner's commands are listed alphabetically in this section. For each keyword, the description includes syntax, comments, examples, and a list of commands to which the keyword can be applied.

after

Designates an MPE/iX command or action to be performed when a specified event occurs. See "Event/Action Mechanism" on page 3-11.

Syntax `after event action`

where *event* is any of the following:

`all` Executes *action* after all files are processed.

`directory` Executes *action* after the RoadRunner directory has been built.

`error` Executes *action* after an error that requires user intervention.

`each [filespec]`
Executes *action* after each file is processed, or after each file matching the *filespec* specification. Wildcards permitted.

`mediaswitch` Executes *action* at each reel change.

`mediavolume[s:]nn`
Executes *action* **after** reel `nn` or after reel *nn* in set `s`. The set number is used with parallel output paths.

where *action* is a quoted string containing any programmatically executable MPE/iX command enclosed in quotes, UDC, or command file. In the case of error events, the action can be `quit`. For example, **after error quit** tells RoadRunner to terminate processing. You can use any of the following system-defined parameters in your MPE/iX command string.

`\errno` The current error number.

`\filename` The fully qualified name of the file currently being processed. This applies only to the **each** and **all** events.

`\time`	The date and time at which the **after** keyword was executed.
`\volid`	The ANSI volume ID of the current reel.
`\volno`	The number of the current reel.

Use a backslash to introduce variables within an MPE/iX command. To use double quote marks within an MPE/iX command, place a backslash before each double quote.

`\"`	Designates a quote.
`\0nnn`	Designates an unprintable ASCII character of value octal nnn.
`\nnn`	Designates an unprintable ASCII character of value decimal nnn.
`\0xnnn`	Designates an unprintable ASCII character of value hex nnn.
`\parm`	Designates a system-defined parameter (listed above).
`\`	When not followed by a digit or variable name, a backslash is interpreted literally.
`\\`	Designates a backslash.

Comments

Only the **each** and **all** events can be specified locally (in relation to a selected set of files) as well as globally (in relation to all files selected in the command). Other events can be specified only globally, once in a command.

In order to stream a job from within a RoadRunner jobstream, your job limit must be set higher than 1, unless the secondary job logs on with the `HIPRI` option.

In the case of the **after all** and **after each** options, files are considered processed once their store bits have been reset. The setting of the **unlock attend | byfile** keyword (which controls whether store bits are reset at the end of each media volume or after each file) affects when the specified actions are performed.

Whenever you interrupt an operation to execute an MPE/iX command, there is a possible negative impact on performance. The effect is minimal for streamed jobs or low-overhead `:RUN` commands, but becomes more apparent if the number of interruptions increase or if CPU-intensive programs are run interactively. It is recommended that you stream jobs to perform the desired actions whenever possible.

Examples

The command below does the following:

- Backs up the prod and test accounts to tapes mounted on three drives, accessed sequentially.
- After each file is backed up, releases it to users (unlock byfile).
- When the prod account is complete, notifies the user mgr.prod.
- Once the third tape has been written, notifies manager.sys that tape mounts are required (the tapes on all three drives have been written).
- After all files have been backed up, notifies manager.sys.

```
select (@.@.prod after all "tell mgr.prod Prod acct ready")
select @.@.test
unlock byfile
after mediavolume 3 "tell manager.sys Mount new tapes!"
after all "tell manager.sys Backup is complete"
store to 17, 18, 19
```

The next example displays an inverse video message followed by a beep on the console after the directory build:

```
full to tape
after directory "tellop The directory build is complete!"
```

The next example streams each file in the group "jobs" whose file name begins with"j" after it is stored.

```
select @.@.myacct
store to tape
after each j@.jobs.myacct "stream \filename"
```

Used with

copy	full	incremental	interim
partial	reload	restore	scan
store	validate		

append

Writes an additional backup to a DAT cartridge that already contains one or more backups. See "Appending Multiple Backups on DAT Cartridges" on page 2-13.

Syntax

append [*nn* | **name** *backupname* | **overwrite**]

where *nn* is a number from 1 to 512 and name is any string up to 35 characters in length that identifies the backup being written.

nn or **name** appends the backup at the specified position in an existing series of backups. If no index number or **name** is specified, the backup is written after all existing backups.

Comments

The **append** keyword tells RoadRunner to write the backup after a previous store on DAT cartridge. RoadRunner automatically numbers each appended backup sequentially and stores this index number in the backup header. You should keep track of the index numbers by using the **report (with header)** option, as shown in the examples below.

If you have a cartridge with multiple backups on it and you store to it using **append** with no index number or name, the new backup is automatically placed after the last existing backup. If you specify an index number or name, RoadRunner overwrites the backup with that number or name and any that come after it. For example, if you specify append 2, RoadRunner overwrites any backups appended after the first. This capability can be useful when performing a series of partial backups after a full backup. Since each partial is cumulative, including backups of all the files stored in prior ones, you can specify **append** 2 to overwrite the previous partial.

The **append** keyword can be used in conjunction with sequential device pools (to tape1, tape2). When the cartridge in the first drive fills up, the next is used. This facilitates unattended backup when appending to an almost full cartridge.

When using the **append** keyword, keep the following in mind:

1. The **append** keyword cannot be used with parallel device paths (to tape1 and tape2).
2. The **append** keyword cannot be used with remote drives.
3. If a backup was created on a remote DAT drive, you cannot append to it on a local system.

Including the **overwrite** parameter with the **append** keyword causes RoadRunner to write the first backup on a tape that is not recognized as a valid RoadRunner tape. If the tape is recognized as a RoadRunner tape, normal **append** functionality is used. If the tape is not recognized as a valid RoadRunner tape, the backup is written to the beginning of the tape, overwriting any existing information. If the **append** keyword is specified without this parameter a valid RoadRunner tape must be mounted.

append can only be specified globally, once in a command.

Examples

To append a backup to a DAT cartridge, sending the header information that contains the index number to a disc file, do the following. (This example assumes the file headlist already exists.)

```
mpe "file headlist,old;acc=append"
incremental to dat append
report (with header, nofiles, norejects to *headlist)
report
```

In the following example, the fifth backup on one cartridge is copied after the first backup on another. Any existing backups after the first one on the destination tape become inaccessible.

```
copy from dat index 5 to dat append 2
```

Used with

```
full   incremental  interim  partial
store  copy
```

autoeject

Ejects DAT cartridges after a RoadRunner operation.

Syntax `autoeject`

Comments By ejecting a DAT cartridge after RoadRunner is finished with it, **autoeject** prevents the cartridge from being written to by any subsequent operation that puts the drive online automatically. It can be used only with local DAT drives (SCSI or HP-IB connections).

autoeject produces an error used with tape units other than DAT drives. The error occurs at the time RoadRunner attempts to eject the first reel.

You can specify **autoeject** only globally, once in a command.

Example The following command might be used for unattended backup to a DAT cartridge. It loads the cartridge automatically and ejects it when the operation is complete.

```
full to DAT autoload autoeject
```

Used with

`copy`	`listdir`	`listinfo`	`store`
`restore`	`validate`		

autoload

Places DAT drives and HP 7979 and 7980 tape drives online without operator intervention.

Syntax

autoload [*pausetime*]

where *pausetime* is the amount of time in seconds that RoadRunner waits to access the tape drive after initiating **autoload**.

Comments

When you use **autoload**, you must specify the logical device number of the device(s) to be put online as a parameter of the **to** or **from** keywords, or use a logical device number in your file equation. If you indicate the device using a logical device class, an error is produced.

Use the `pausetime` parameter to accommodate the time it takes the device in use to load. The default *pausetime* is 30 seconds. While reel-to-reel tape drives take about 10 seconds to load, DAT drives can take anywhere from 30 to 120 seconds to load. If the *pausetime* is too low, RoadRunner generates a console request asking if the tape is really on the specified device:

```
IS "FILENAME" ON LDEV#nn (Y/N)?
```

If an error occurs during the attempt to place the drive online, RoadRunner reports the system error number. Such errors do not always mean that the drive was not put online successfully. RoadRunner produces non-fatal errors to help in the identification of persistent equipment problems.

The **autoload** keyword is particularly useful for sites with DAT drives, which do not come back online automatically after a power failure. As a further aid to unattended operations, use the **autoreply** keyword to convert your tape drive to autoreply mode so an operator need not be present to enter a `:REPLY` command. See the **autoreply** keyword on page 5-8.

You can specify **autoload** only globally, once in a command.

Example

The following command automatically places logical device number 7 online, waits 120 seconds before doing the **autoreply**, and backs up all files on the system:

```
full to 7 autoload 120 autoreply
```

Used with

`copy`	`full`	`incremental`	`interim`
`listdir`	`listinfo`	`partial`	`reload`
`restore`	`store`	`validate`	

autoreply

Changes tape drive(s) from manual reply mode to automatic reply mode, and back to manual reply mode once the backup is complete. Useful when performing unattended backups.

Syntax

```
autoreply
```

Comments

When you use **autoreply**, you must specify the logical device number of the device(s) as a parameter of the **to** or **from** keywords, or use a logical device number in your file equation. If you indicate the device using a logical device class, MPE iX generates a tape reply request on the console.

If you specify **autoreply** in a command with multiple device paths, RoadRunner changes them all to autoreply mode for the duration of the operation.

If the specified device was already configured for auto reply, RoadRunner does nothing, and leaves the device in autoreply mode after completing the operation.

The **autoreply** keyword is not supported when backing up to devices on remote systems. To configure a remote device for autoreply, do one of the following:

- Use the MPE/iX SYSGEN utility to configure the remote system device for auto reply.
- Use the AUTOREP utility supplied with RoadRunner on the remote system. See "AUTOREP," on page B-1.

autoreply can only be specified globally, once in a command.

Example

In the following command, RoadRunner automatically places logical devices numbered 7 and 8 online, and configures them for auto reply until the unattended backup is complete. Logical devices 7 and 8 are then re-configured to normal reply mode.

```
full to 7, 8 buffer autoload 60 autoreply
```

Used with

copy	full	incremental	interim
listdir	partial	reload	restore
scan	store	validate	

backupname

Names an appended backup. See "Appending Multiple Backups on DAT Cartridges" on page 2-13.

Syntax

backupname *"name"*

where *name* is any string up to 35 characters in length that identifies the backup being written.

Comments

backupname can be specified only globally, once in a command. This keyword can be used with the **append** command to assign names to backups, or overwrite existing backups by name. The **backupname** keyword can be used with the **index** command to read backups by name. The backupname assigned to this backup will be used by subsequent read operations (indexname) or write operations (appendname).

The **backupname** keyword is case sensitive, as are the commands that use the assigned name.

Examples

In the following example, the files in the testdata group testdata are written to the tape as backup number 2, and are assigned the name testdata. This data can later be selected using either its backup number (2) or backup name (testdata).

```
<01>store
<02>to 7
<03>select @.testdata
<04>backupname "testdata"
<05>/
<rr>go
...
This is appended backup number 2, beginning on volume 1
```

In the following example, a copy of the backup named testdata overwrites the backup named second, and is given the name new second. The **check** command has also been used in this example to check the syntax of the command string before it is executed.

```
<01>copy
<02>from 7
<03>to 9
<04>indexname "testdata"
<05>appendname "second"
<06>backupname "new second"
<07>/check
No syntax errors were found in the current command
<rr>go
...
This is appended backup number 2, beginning on volume 1
RoadRunner starting to copy files to media volume 1
RoadRunner finished writing to media volume 1 on device 19
278 files (73344 compressed sectors) written to this volume
278 files were copied
```

Used with

copy	full	incremental	interim
partial	store		

change

Alters file, creator, group, account names and path. The `path` modifier allows a path or set of paths to be changed on either a global or local basis when storing or restoring files.

Syntax

```
change file|group|account|creator|path oldname to newname
[,...]
change path frompathspec to topathspec
```

where *frompathspec* is an HFS-compliant path specification and *topathspec* is either an HFS compliant path name or a fully-qualified MPE filename.

where you can use any wildcard character in *oldname*, but only the `@` wildcard in *newname*. The *oldname* and *newname* must contain the same number of wildcard characters. For a list of wildcards, see "Wildcard Characters in Fileset Specifications" on page 2-23.

Comments

change can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files; enclosed within the parentheses following the **select**). If you use `change` globally, it executes changes on all files selected for the operation that conform to the change criteria. However, RoadRunner allows only one change per level, per file. If a file's group is altered in a **local change** statement, it will not be altered again by a **global change** statement. See the Examples below.

When **change** is used, any report produced by the operation reflects the changed names.

If the **change** keyword is specified with the **create usedir** option, RoadRunner first looks in the backup directory to determine the characteristics of the entities. If no directories were backed up, it looks in the target system's directory for the entities. If these exist, the created entities inherit their characteristics. Otherwise, they are created with default characteristics.

Change Creator Modifications for MPE 5.0 and Later

MPE/iX 5.0 allows users to create files in accounts other than the user's home account. This capability requires the operating system to track both the creator's username and the creator's accountname in the file label of all new files created by MPE/iX 5.0 and greater. Files which were created before updating to MPE/iX 5.0 will still have a blank accountname portion of the creator name. MPE and RoadRunner understand this to imply that the creator is in the same account as the file. This change has several impacts on the manner in which RoadRunner operates.

When changing the account of a given file with the **change account** keyword, the user's account in the file label is not changed. For example, files A ... Z exist in group B of account C and are created by U1 ... U26, respectively. These files are backed up as follows:

```
select @.B.C store to 7
```

and restored:

```
select @.B.C restore from 7 change account C to X
```

All files restored will still be created by U1 ... U26 in account C, for example file A.B.X will have a creator of U.C.

If the desired effect is to change the creator of these files to reside in the new account, (i.e. the new user from the example would be `U@.X`) the **change creator** keyword must also be applied. The **change creator** syntax has been modified in RoadRunner 4.5 to allow for this specification, and is as follows:

```
change creator olduser[.oldacct] [to] newuser[.newacct]
```

If either *oldacct* or *newacct* is specified, the other must be specified as well. If accounts are not specified, only the user portion of the creator will be changed.

```
change creator a to b.c *** invalid ***
change creator a.b to c *** invalid ***
change creator a to b      [account name in creator unchanged]
change creator a.b to c.d [change user and account names in creator]
```

To change users U1 ... U26 to be in account X rather than account C, the following change must be applied:

```
change creator @.C to @.X
```

If the **create** keyword is used, and these changed creators do not exist, they will be created if the accounts associated with those users exist. In the event that an account does not exist (either because the original account of the user is nonexistent and **no change creator** was specified or because the creator's account was changed to something that does not exist), the creator and associated account will be placed in the file label as previously described, but the user will not be created and a warning will be issued:

```
RoadRunner WARNING ###: Creator U.C not created because account
C does not exist
```

This change also has impacts on files that are being migrated between pre-4.5 and post-4.5 systems:

1. Any file restored by RoadRunner to a 4.5 or greater system that originally did not have a creator's account in the file label will be restored without the account portion.
2. Any file restored by RoadRunner on a pre-4.5 system that did have a creator account will result in there being no creator account on the restored file.

Examples

The following command restores all files in the prod account, changing the account name from prod to oldprod. Since **create usedir** is specified, RoadRunner accesses the MPE/iX directory to transfer the appropriate attributes to the created entities. If no directory information was stored on the backup, it looks for a prod account in the directory on the target system and replicates its attributes.

```
select @.@.prod
restore from 7
change account prod to oldprod
create usedir
```

In the next example, files in the prod account whose names begin with K followed by seven digits are changed to names starting with Q followed by the same digits.

```
select @.@.prod
restore from 7
change file K####### to Q@@@@@@@
```

Note that the number of wildcard characters in the original name must match the number used in the new name.

The following example performs multiple changes on the prod account. All account names are changed from prod to test; the owner is changed from mgr.acct to mgr.test; all files in the source group are changed to the srcnew group; and any groups whose names begin with p are changed to ones beginning with t.

```
select (@.@.prod change acct @ to test, creator mgr.scct to
mgr.test, group source to srcnew, group p@ to t@)
restore from 7
```

The command below stores the parts, inv, and ledger accounts. The account identifications of files in the parts and inv accounts are changed to myparts and myinv. The files in the ledger account are unchanged.

```
select (@.@.parts, @.@.inv change account @ to my@)
select @.@.ledger
store to 17
```

Note

When changing two accounts to one new account, if there are duplicate file names, the file from the second selected account would overwrite the first file.

When a file meets both local and global change criteria, only the local criteria affect it. In the next example, the first letter of each file beginning with a is changed to c except those in the @.pub.sys account. Files in pub.sys beginning with a are changed to begin with b.

```
select (@.pub.sys change file a@ to b@)
select /
change file a@ to c@
```

The next example further illustrates the interaction of local and global criteria.

```
select (@.@.myacct change group xyz to abc)
select (@.@.@ excluding @.@.myacct)
change x@ to y@
```

In the preceding example, if myacct consisted of the groups xyz, xxx, and pub, the results would be as follows:

xyz - Changed to abc by local change
xxx - Changed to yxx by the global change; didn't match the local one
pub - Unchanged; matches neither local nor global criteria

To prevent a global change from affecting a selected fileset, specify overriding local change criteria for that fileset. In the following example, the group and account names of files in myacct are not affected by the global change, since they are changed to themselves by local criteria. myacct is excluded from the second **select** statement only to prevent RoadRunner from trying to store it twice; the exclusion does not affect the global change from being applied. The local change is what preserves the filenames.

```
select (@.@.myacct change group @ to @, acct @ to @)
select (@.@.@ excluding @.@.myacct)
change group xb@ to zb@, acct apr@ to may@
```

Used with

full	incremental	interim	partial
store	reload	restore	

Comments

The ability to change the path on store or restore is especially useful when moving files between two machines with different directory structures, or to a new directory location on the same machine.

When specifying path changes, the presence of a slash at the end of the pathname determines whether the change effects files recursively or non-recursively. An additional

restriction is that if the *frompathspec* ends in a slash (recursive) then the *topathspec* must also end in a slash. For example:

```
change path /proj1/usr1/ to /proj1/usr2/
```

moves all files that are descendants of /proj1/usr1 (recursively) to the directory /proj1/usr2. The effect of the above command is shown on several hypothetical files in the table below:

Original Name	Changed Name
/proj1/usr1/source.c	/proj1/usr2/source.c
/proj1/usr1/a.out	/proj1/usr2/a.out
/proj1/usr1/gif/mountain.gif	/proj1/usr2/gif/mountain.gif
/proj1/usr1/gif/misc/sol.gif	/proj1/usr2/gif/misc/sol.gif

However, if you specify:

```
change path /proj1/usr1 to /proj1/usr2
```

Only those files in the /proj1/usr1 directory (or the file /proj1/usr1, if it is a file) are changed.

The use of slashes to initiate a recursive file selection must be consistent in the *frompathspec* and the *topathspec*. For example, specifying either of the following yields a syntax error:

```
change path /proj1/usr1/ to /proj1/usr2
change path /proj1/usr1 to /proj1/usr2/
```

The one exception to this rule is when using MPE syntax in the *topathspec* to force the restored files/directories into MPE namespace. The syntax used in the *topathspec* determines whether the files are restored as part of **MPE namespace** (in a group) or **POSIX namespace** (in a hierarchical directory)

To change the path of a file so that it remains in a hierarchical directory, specify a HFS pathname as the *topathspec*, as in the pervious examples. This ensures that the files are restored into POSIX namespace, **unless** there is an existing MPE group on the target directory with the same first two components of the pathname in the `topathspec`.

For example, the following command restores the /proj1/usr1 POSIX files to an existing MPE group PUB in the account DEV (PUB.DEV):

```
change path /proj1/usr1 to /DEV/PUB
```

If the /DEV/PUB account and group do not exist on the target directory, the /proj1/usr1 files are restored to the POSIX directory /DEV/PUB.

To force files into MPE namespace, use MPE syntax in the *topathspec*. For example, to force the /proj1/usr1 files into the PUB group in the DEV account, enter the following command:

```
change path /proj1/usr1 to PUB.DEV
```

Most often, the result of a **change path** is an HFS directory descendant from the root. The following instances are exceptions.

- When the *topathspec* lies within an existing MPE group, the result is a POSIX directory descendant from the MPE group.

For example, given the following directory structure:

/usr/include/tom

change path /usr/include/ to /Admin/Mgrs/usr/include/ *yields* **/Admin/Mgrs/usr/include/tom**

- When the *topathspec* is in MPE syntax, the resultant directory becomes part of MPE namespace.

  For example, given the following directory structure:

  /usr/include/gear

  change path /usr/include/ to MISC.PROD *yields*
  /PROD/MISC/usr/include/gear *in HFS syntax*

When **change path** is used, RoadRunner does not preserve attributes such as Group Identifier (GID) and User Identifier (UID). This is because there is no natural correspondence between the original path name and the new path name.

In the following examples, RoadRunner will not preserve GID and UID attributes because there is no corresponding level for the directories:

```
change path /A/ to /X/Y/Z/
change path /A/%/X to /X/Y/%.
```

Using Wildcards with the Change Keyword:

Wildcard use with the change keyword follows RoadRunner's file selection specifications, with the following added restrictions:

- The @ sign is the only valid *topathspec* wildcard, except for the % sign.
- You cannot specify consecutive % signs such as /%/%/%/X to /%/%/X
- The number of % wildcards used in the *topathspec* must be less than or equal to the number in the *frompathspec*. For example, given the following path:

  /one/two/three/x

  change path /%/x to /a/%/x *yields*
  /a/one/two/three/x

  and

  change path /%/x to /a/x *yields*
  /a/x

  but

  change path /%/x to /%/a/%/x *yields an* ERROR

  because the second % does not match anything in the *frompathspec*.

- Wildcard symbols other than % must reside in corresponding *levels* of the path. For example, given the following path:

  /abchello/one/two/dir/x

  change path /abc@/%/dir/x to /xyz@/%/x *yields*
  /xyzhello/one/two/x

  but

  change path /abc/def@/%/x to /xyz@/%/x *yields an* ERROR

because the @ wildcard in the xyz directory of the *topathspec* has no corresponding wildcard in the *frompathspec*.

- Each time you use the % wildcard, RoadRunner resets the component level. The component level is the location of an object (file or directory) in the pathname, expressed in places from the root. For example, given the following path:

  abc/one/two/three/xyzold/dir/x

  `change path /abc/%/xyz@/dir/x to /a/b/c/%/new@/newfile` *yields*
  `/a/b/c/one/two/three/newold/newfile`

  because the xyz@ and new@ directories fall in the same place relative to the % wildcard, but if the another directory is placed between the two in the *topathspec*:

  `change path /abc/%/xyz@/dir/x to /first/%/another/new@/newfile`

  an ERROR results, because the new@ directory no longer lies in the same place in relation to the % wildcard directory.

- Wildcards can be specified (for matching purposes) in the *frompathspec* where there is no corresponding level in the *topathspec*. In this instance, the wildcards are used for matching purposes only, not to replace any characters or objects. For example:

  `/FIRST@/SECOND@/THIRD@/ to /ONE@/`

- Wildcards can be specified to eliminate intervening directories. In the following example, the gamma directory is eliminated:

  `change path /alpha/beta/gamma/%/@ to /alpha/beta/%/@`

combine

Pro Module

Merges source sets, and outputs combined destination set(s). Used only with the **copy** command, when copying from a backup composed of multiple tape sets created in parallel.

Syntax

```
combine
```

Comments

Regardless of the number of tape drives used to create a parallel backup, only two tape drives are needed to perform a copy. RoadRunner preserves the sets that comprise the backup, copying files from set 1 to set 1, files from set 2 to set 2, etc. However, if you are only copying a few files from each set, or are copying from a low-capacity reel-to-reel tape to a high-capacity DAT, this is inefficient in terms of media usage.

When you use **combine**, RoadRunner merges the input sets. If you are using only one output device, RoadRunner merges all input sets into a single set. If three or more parallel "Input Sets" are to be combined into two or more parallel "Output Sets," the number of "From" Ldevs separated by "and" must equal the number of "To" Ldevs separated by "and".

The **combine** keyword can only be specified globally, once in a command.

Examples

The following command copies a backup with multiple sets created in parallel from a reel-to-reel device in class `tape` to a `DAT` device. All sets in the source backup are combined into one set in the destination.

```
copy from (tape name src1) to (dat name dest1) combine
```

Four parallel path input set from the backup combined into a two path parallel output set using 6 Ldevs:

```
copy from 7, 8 and 9, 10 to 11 and 12 combine
```

From the four parallel backup, the 4 input paths will be loaded on the following ldevs:

```
path 1 on ldev 7, path 2 on ldev 8, path 3, on ldev 9, path 4 on
ldev 10.
```

The result would be:

```
input path 1(Ldev 7), input path 2(Ldev 8) combines to output
path 1 on Ldev 11; input path 3(Ldev 9), input path 4(Ldev 10)
combines to output path 2 on Ldev 12
```

Used with

`copy`

compress

Sets the level of data compression in a RoadRunner backup.

Syntax

compress [*n*]

where *n* is one of the following:

0 no compression

1 default compression

Pro Module

2 high compression (Pro module only)

Comments

The exact ratio of data compression varies according to the nature of the files being compressed. Due to the added processing time, higher compression takes more CPU time than lower compression. The trade-off depends on your backup objectives and the power of the CPU.

compress can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Examples

The following command selects files in the dallas and dev accounts, compresses them using high compression, and stores them to logical device 17:

```
select @.@.dallas, @.@.dev
compress 2
store to 17
```

The following command compresses files in the dallas account with high compression and files in the chicago and dev accounts with default compression.

```
select @.@.chicago
select (@.@.dallas compress 2)
select @.@.dev
compress 1
store to 17
```

Used with

full	incremental	interim	
store	partial		

concurrent

Prints RoadRunner reports as files are processed.

Syntax `concurrent`

Comments When you use the **concurrent** keyword, RoadRunner reports on files as they are processed. It sends report information to the listing device immediately (to your terminal, if you are running interactively). The added processing time can cause operations to run slower than usual.

The **concurrent** keyword can only be specified globally, once in a command, and applies to all **report** statements.

Examples The following command generates a report of the name and filecode of each file in the pub account as it is processed:

```
select @.@.dallas
report (fullname, filecode)
concurrent
store to 17
```

Used with

`copy`	`full`	`incremental`	`interim`
`partial`	`reload`	`restore`	`store`
`validate`			

copyacd/nocopyacd

When **copyacd** is specified, RoadRunner stores or restores ACDs (Access Control Definitions) for the files processed. **nocopyacd** disables this command. **copyacd** is the default with the following exception, affecting store operations only: if the user does not have SM, OP, or RACD capability, ACDs are not copied.

Syntax

```
copyacd
nocopyacd
```

Comments

Use **nocopyacd** when you want to store files without their ACDs—for example, when creating a tape for distribution to another site with different security.

copyacd and **nocopyacd** can be specified globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

The following command stores files in the pub.sys group with ACDs. ACDs for all other files in the mfg account are not stored. **nocopyacd** is specified globally, and **copyacd** is specified to override it locally for the first **select**.

```
select (@.pub.sys copyacd)
select @.@.mfg
nocopyacd
store to 17
```

Used with

full	incremental	interim	partial
reload	restore	store	

create

Builds new accounts, groups and users for restored files if they do not already exist on the destination system.

Syntax

create [*optionlist*]

where *optionlist* is one or more of the following, separated by commas:

all	Creates groups, accounts, creators, and paths as applicable. This is the default option.
group	Creates groups only.
account	Creates accounts only.
creator	Creates users only.
path	Creates path, group, and account information. This option creates the directory tree for MPE or POSIX directory and file structures.
usedir	Uses any relevant directories on the backup to assign capabilities and privileges to created users, groups and accounts. This option overrides the default capabilities.

Comments

Where *group*, *account* and *creator* are specified, **only** that element is created. If you specify **create account** without also specifying **group**, and the necessary group does not exist on the destination system, the account is **not** created.

The *account*, *group*, or *users* created are only those that are referenced in the label of the files being restored, all others will not be created. Use the **directory** keyword for that.

If the **usedir** option is not specified with **create**, or if **usedir** is specified but the MPE iX directory does not exist on the backup, RoadRunner assigns default capabilities to any users, groups and accounts created.

If the **usedir** option is specified and there is no directory information on the backup, RoadRunner creates new entities with default characteristics. The one exception to this is when you specify the **change** keyword. In this case, RoadRunner looks in the target system's directory for the original entities. If these exist, the newly created entities will inherit their characteristics. If not, RoadRunner creates them with default characteristics.

RoadRunner only creates users who are creators of the files that are being restored.

create can be specified globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify **create** locally, place it within the parentheses following the **select**.

Examples

The following command restores all files from the backup on logical device 17, creating any accounts or groups that do not exist.

```
select @.@.@ restore from 17
create account, group
```

The following command restores all files from the backup on logical device 17, creating any accounts or groups that did not exist for the dallas account only.

```
select (@.@.dallas create account, group)
select @.@.@
restore from 17
```

The following command restores all files in the /proj/usr1/ directory, creating the *path*, *group*, and *account* information that did not exist.

```
select /proj/usr1/ create path
restore from 17
```

The following command restores the prod and dev accounts, creating any necessary *users*, *groups*, and *accounts* with information from the MPE/iX directory on the backup.

```
select @.@.prod, @.@.dev
restore from 17
create usedir
```

Used with

`reload` `restore`

datefile

Specifies an alternate file in which to store and retrieve full/partial/interim backup dates, allowing you to keep multiple schedules automatically.

Syntax `datefile` *filename*

Comments RoadRunner's backup macros -- **full, partial, incremental** and **interim** —select files according to whether their modification date is greater than the backup date in the bpfbdate.pub.sys file. The **datefile** keyword tells these macros to use an alternate file for backup date flags. This allows you to keep multiple schedules automatically.

For instance, some sites can't perform a weekly full backup all at once because of time constraints, so they stagger full backups of selected accounts or applications throughout the week. Each of these "sub-backups" could have its backup dates stored in a separate date file, thereby automating a potentially complicated task.

The date file you specify is used exactly as bpfbdate.pub.sys. For each level of backup, there is a corresponding date flag that is checked and reset to the current system date and time. For example, to perform a full backup, RoadRunner backs up all files and resets all backup dates in the date file. To perform a partial backup, RoadRunner checks the full backup date, selects all files modified since that date, and resets the partial date flag to the current system date and time. The following table shows each type of backup with the date flags it checks and sets.

Backup Type	Checks	Sets
`full`	No dates	Full, partial, and interim backup date
`partial`	Full backup date	Partial and interim backup date
`interim`	Partial backup date	Interim backup date
`incremental`	Interim backup date	Interim backup date

You can also set the date flag in any date file manually with the **fulldate, partdate**, and **interimdate** commands.

Examples The following command backs up the acctng account and sets the full backup date in the file actdate.pub.acctng to the current system date and time.

```
select @.@.acctng
full to tape
datefile actdate.pub.acctng
```

The following command backs up files in the acctng account modified since the last full backup of the acctng account according to the date and time recorded in actdate.pub.sys.

```
select @.@.acctng
partial to tape
datefile actdate.pub.sys
```

The following sets the partial backup date in devdates.pub.dev to 03/01/96.

```
partdate 03/01/96 datefile devdates.pub.dev
```

The following command backs up files in pub.sys that have been modified since four days after the full backup date in the file sysdates in your logon group.

```
select (@.pub.sys where moddate >= lastfulldate+4)
store to tape
datefile sysdates
```

Used with

full	incremental	interim	partial
store	fulldate	partdate	interimdate

dbfast

Eliminates the test for data above the high water mark when backing up IMAGE datasets.

Syntax `dbfast`

Comments The high water mark is a pointer to the highest active entry in a dataset. However, it is possible for a detail dataset to have a corrupted delete chain, causing new entries to be placed between the high water mark and the end of the file.

When backing up a database, RoadRunner's default is to check for data above the high water mark. This ensures that even if this error has occurred, all data will be stored. If you're sure that the database is structurally sound, you can save time when backing up the database by using the **dbfast** keyword to eliminate this check.

dbfast can be specified globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify **dbfast** locally, place it within the parentheses following the **select**.

Examples The following command backs up databases in the dallas account without checking for data above the high water mark.

```
select @.@.dallas
dbstore dbfast
store to 17
```

Used with

`full` `incremental` `interim` `partial`
`store`

dbrestore/ nodbrestore

When **dbrestore** is specified, RoadRunner performs several commands to ensure the integrity of a restored IMAGE database. The default, **nodbrestore**, disables these functions.

Syntax

```
dbrestore
nodbrestore
```

To maintain the integrity and synchronization of restored databases, RoadRunner does the following when **dbrestore** is specified:

- Restores all datasets in a database if the root file is selected.
- Does not restore a database that was not stored with the **dbstore** keyword, as this is the only way it can ensure that the backed up version of the database is complete.
- Prevents access to the destination database during restore, so that access to the disc version of the root file cannot prevent the backup version from being restored.

Note

Cannot be used with parallel paths. RoadRunner needs to restore and lock the root file before restoring sets, and does not handle the interlocks between paths at this time.

dbrestore and **nodbrestore** can be specified globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

In the example below, **dbrestore** is used when restoring databases in the `parts` account.

```
select @.@.parts dbrestore restore from 17
```

Used with

`reload` `restore`

dbstore/nodbstore

When **dbstore** is specified, RoadRunner performs several functions to ensure that databases are stored in a manner compatible with HP's automated transaction logging and recovery facility. **nodbstore**, the default, disables those functions.

Syntax

```
dbstore
nodbstore
```

Comments

To maintain the integrity and synchronization of the stored databases, RoadRunner does the following when **dbstore** is specified:

- Stores all datasets in the database if the root file can be stored.
- **Online Module** If **online** was not specified (online module only), it prevents write access to the destination database during store, so that access to the disc version of a file cannot prevent any dataset from being stored. If the **nolock** option is specified, **dbstore** overrides it for databases only.
- Does not store the database if the root file cannot be stored.
- Resets the date and time stamp and the dirty bit in the root file in the manner required for automated transaction logging and recovery.

If **dbstore** and **nolock** are both specified, **dbstore** overrides **nolock** for databases only, preventing write access to database files.

dbstore and **nodbstore** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

In the example below, **dbstore** is used to store the databases in the parts account.

```
select @.@.parts dbstore store to 17
```

Used with

```
full   incremental   interim  partial
store
```

description

Specifies descriptive text that is written to the backup header.

Syntax **description** *string*

where *string* is any text up to 255 characters.

The **description** keyword can be used, whether or not you also use tape labels, to provide additional descriptive information about the backup. BackPack/XL users who are accustomed to specifying descriptive comments in the *string* portion of the BackPack tape label can use this keyword for that purpose.

You can print the contents of the backup header, shown on page 5-58, with the **report** keyword.

description can only be specified globally, once in a command.

Examples In the command below, a **description** is specified.

```
store @.@.acctg to 7
description "This is a checkpoint backup of "
"the accounting system before the GL update."
```

The following lines would then appear in the backup header. To list the backup header, use the **report (nofiles, header)** option. For a complete listing of the contents of the backup header, see "Backup Header" on page 5-58.

```
Backup Description:
This is a checkpoint backup of the accounting system before the
GL update.
```

Used with

store	full	incremental
partial	interim	copy

directory/nodirectory

Controls whether the MPE iX directory structure is included in RoadRunner commands.

Store syntax

```
nodirectory
directory [minimum | all]
```

Note

When using the minimum option, only the user, group, and account that are used in the file label are included in the directory.

When **directory** is specified on a store operation with no options, RoadRunner stores all directory information.

`minimum`	Stores only the directory information (user, group, and account) associated with the files selected to be stored.
`all`	Stores all aspects of the directory structure (user, group, account, and UDCs). This is the default for store operations when directory is specified.

Restore syntax

```
nodirectory
directory [udcs | update | all]
```

When **directory** is specified on a restore operation with no options, RoadRunner creates missing accounts, groups, and users in the system directory as necessary to restore files. Existing entities are not modified unless you specify **update** or **all**.

`udcs`	Restores UDC pointers or references to `COMMAND.PUB.SYS`.
`update`	Alters account structure and capabilities in the system directory to match the directory on the backup. Users, accounts, and groups that exist in the system directory, but not in the MPE/iX directory on the backup, are not disturbed.
`all`	Combines **udcs** and **update** commands.

directory all is the default for the **reload** command. **nodirectory** is the default for the **restore** command. For minimal directory restore use **create usedir.** See page 5-10.

Copy syntax

```
nodirectory
directory
```

Comments

To store only the directory structure without backing up data files, specify **directory** with no **select** on a **store** operation. To rebuild the directory structure without restoring files, specify **directory** with **no select** on a **restore**. When you specify **directory** with no **select** statement, RoadRunner's default file selection is different than usual. Ordinarily, if there is **no select**, @.@.@ or / is the default. But when directory is specified with no **select**, no data files are stored.

If **onvs** or **notonvs** are specified with **directory**, only the directories of those volume sets are included in the store.

Note

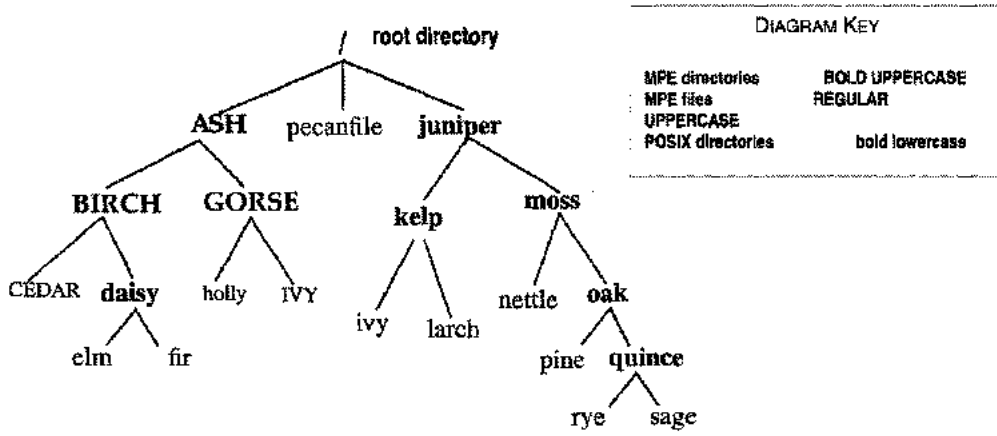
Whenever you perform a store with the **directory** keyword, make sure you also store COMMAND.PUB.SYS. This file describes your system UDC structure. When you restore a particular version of your system directory using the udcs option, you *must* restore the corresponding copy of COMMAND.PUB.SYS to avoid UDC initialization errors. After the restore, log off and log on again to activate the restored UDCs.

directory or **nodirectory** can only be specified globally, once in a command.

There are several things to keep in mind when dealing with POSIX directories:

- All POSIX directories that descend directly from the root (not descendant from MPE directories) are considered to be spanned to MPEXL_SYSTEM_VOLUME_SET. If the system volume set is selected, these directories are also selected.
- All POSIX directories that are descendants from MPE groups are spanned to the volume set from which the parent group is descended.
- When **directory minimum** is specified, RoadRunner selects only those POSIX directories which lie on the path of a specified directory or file.

  For example, if the rye file in the diagram below is selected, only the juniper, moss, and oak directories are included.



Examples

In the command below, the dallas account is stored with directory information pertaining to that account.

```
select @.@.dallas directory minimum store to (tape name t)
```

In the command below, only the directory and UDCs are restored.

```
restore directory all from 7
```

When you create a test environment or move accounts to another system, you can use **directory minimum** to store only the directory information pertaining to the files stored.

```
select @.@.acct1, @.@. acct2 store to 7
directory minimum
```

If you use **directory update** when restoring the accounts, the directory structure for any identically named accounts already existing on the system is modified to match that stored on the tape.

```
restore from 7 directory update
```

If you did not have a store tape with just the accounts you wanted to restore, and were instead restoring them from a full backup, you would use the **create usedir** option to build the accounts with the capabilities on the tape. For more information see the **create** keyword on page 5-20. The **directory** option would not be required.

```
select @.@.acct1, @.@. acct2 restore from 7
create usedir
```

After all directories have been selected, the total number of POSIX directories is printed, as shown in the following example:

```
RoadRunner 4.5.0 * Copyright 1991-96 Unison Corp. * Wed 14Jan96
9:09am
<01>sel /users/ store to (disc name vt) directory
<02>/g
Starting Directory Build
Reading MPE/XL directories
Scanning POSIX root directories
Scanning volume set: MPEXL_SYSTEM_VOLUME_SET
Scanning volume set: USER.VOLUME.SET
MPE/XL directory processing complete
Finished Directory Build
2 MPE/XL directories selected to be stored
122 POSIX directories selected to be stored
14 files selected to be stored
3328 sectors of data to be stored
Storing files to virtual tape fileset VT
Writing to VT file VT.PUB.BACKUP
2 MPE/XL directories stored
122 POSIX directories stored
14 files were stored
3328 uncompressed file sectors were stored
2240 compressed sectors written to the backup media
Compressed data required 67% of uncompressed file space
```

Used with

copy	full	incremental	interim
partial	reload	restore	store

display

Specifies which messages RoadRunner displays and where they are sent.

Syntax

display [*msglist*]
or
display (*msglist* [**to** *destlist*])

where *msglist* is one or more of the following, separated by commas:

all	All messages.
errors	Any problem that requires operator intervention before RoadRunner can continue processing.
normal	Basic status messages.
progress [n]	Progress of operation reported every *n* minutes. The default is 1.
statistics	CPU time, elapsed time, and throughput rates of operation.
warnings	Minor errors that do not interrupt processing.

where *destlist* is one or more of the following, separated by commas:

#Snn	Sends messages to session *nn*.
user.account	Sends messages to a user.
**file*	Sends the message to the back-referenced file.

Comments

The following are the defaults for RoadRunner's messages:

- If the **display** keyword is not used to control messages or if no *msglist* is provided, *normal*, *warning*, and *error* messages are displayed.
- If no *destlist* is specified when you are running interactively, messages are sent only to $STDLIST. In batch mode, messages are sent to $STDLIST and the console.
- If *destlist* is specified, messages are sent to the specified destination and $STDLIST. You cannot send messages to a user or session in QUIET mode.
- If no interval is specified for progress messages, they are displayed every minute.

If an irrecoverable error occurs on a particular volume, requiring that reel or cartridge be discarded and replaced with another, the messages reporting sector count and percent completion will be inaccurate. This is because some files have been written twice, once to the discarded reel and once to the replacement reel. Due to the nature of progress reporting, RoadRunner is unable to reset the progress percentage; completion is eventually shown as greater than 100%. This is normal in the situation described. See "Store Errors" on page 3-28 for more information.

display cannot be specified locally. Though it can be used more than once in a command, its application is always global.

Example

The following command prints error messages and sends progress reports every five minutes to session #15, the line printer, and $STDLIST.

```
full to tape
display (errors, progress=5 to #s15, *lp)
```

Used with

copy	full	incremental	interim	scan	store
listdir	partial	reload	restore	validate	

excluding

Excludes files from RoadRunner commands.

Syntax

excluding *fileset* [,*fileset*]...
or
excluding (*fileset* [,*fileset*]... **where** *expression*)

where *fileset* is any of the following:

filename The file or files to be processed; can contain wildcards and follows standard MPE/iX naming conventions.

^indirectfile

A text file listing fully-qualified file names (wildcards permitted) delimited by blanks or carriage returns. Comments, delimited by braces ({ and }) are allowed anywhere in the file and can span multiple lines.

where the syntax for **expression** is described under the **where** keyword on page 5-78.

Comments

You can use any number of **excluding** statements in a command, but there can be only one **where** statement per **excluding** keyword.

Though indirectfiles are supported for the convenience of users migrating from other backup software, RoadRunner's include file capability provides greater flexibility and better performance. Please see the discussion on page A-5. The **include** keyword is covered on page 5-38.

excluding can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Examples

The following statement will store the SYS account first, followed by the rest of the accounts, and print a store listing on a LaserJet printer using a 4up landscape format.

Note

The parenthesis (and) in the second statement must be included for this statement to function properly.

```
select @.@.sys
select (@.@.@ excluding @.@.sys)
store to 7 autoreply
report (to 25 format 4up, landscape)
```

The following statement selects all files in the prod account, except those in the pub group, and all files in the dev account.

```
select (@.@.prod excluding @.pub.prod)
select @.@.dev
```

The following statement selects the files listed in an indirectfile called myaccts, except those in an indirectfile called nostore.

```
select (^myaccts excluding ^nostore)
```

Used with

`copy`	`full`	`incremental`	`interim`
`listdir`	`partial`	`reload`	`restore`
`scan`	`store`	`validate`	

from

Specifies the device or virtual tape file where the files to be read are found.

Syntax

from *sourcelist* **[and** *sourcelist***]...**

where *sourcelist* is one or more of the following items, separated by commas. Devices in a *sourcelist* are read from sequentially.

******file*	Back-references an MPE/iX file equation.
devdescr	A logical device number or class, plus optional file name and user label information.

Pro Module

where **and** joins devices to be read from in parallel (Pro module only).

where *devdescr* is one of the following:

dev#
or
devclass
or
(*dev#*|*devclass* **[name** *name***][label [ansi|rr]][volid [***volid***]]**)

where:

name	The name of the tape file or virtual tape file. If this option is used, no file equation is required.
label	The type of label, either **ansi** or **rr**. **ansi** is the default.
volid	The volume ID. Up to 6 characters.

Comments

To reduce retrieval time, RoadRunner can read from multiple devices, either sequentially or in parallel. The devices in a sequential pool are read from one at a time. When one reel completes, RoadRunner begins reading from the next device defined in the *sourcelist*. This eliminates the time required to rewind the tape and mount the next reel. When listing devices to be read from sequentially, separate them by commas.

When multiple devices are to be read from sequentially, only the first device in the pool can be qualified with label and density information. Subsequent devices inherit these characteristics.

When multiple devices are to be written to sequentially, only the first device in the pool can be qualified with label and density information. Subsequent devices inherit these characteristics.

The devices defined in a parallel pool are read from concurrently. To read from devices in parallel, the source tape must have been also backed up in parallel. Reading from tapes in parallel can improve performance when the processing speed of your CPU outstrips the transfer rate of slow input devices like DAT drives. When listing devices to be read from in parallel, separate them by the word **and**. If your are only restoring a small portion of the backup, a sequential restore is recommended.

If the **label option** is not specified, but a **volid** is specified, or if **label** is specified without a label type, ANSI labeling is assumed. If **label** is specified but the **volid** option is not, RoadRunner prompts you for the volume ID. However, if a tape library management system is in use, it automatically supplies a volume ID. See "Labeled Tapes" on page 3-13., for details.

When reading from virtual tape files, **file* must reference the name of the root file.

from can only be specified globally, once in a command.

Examples

To read from a single device defined in a file equation:

```
from *t
```

To read from three devices sequentially, all defined by file equations:

```
from *t1, *t2, *t3
```

To read from two devices, defined by file equations, in parallel:

```
from *t1 and *t2
```

ANSI labelling is assumed when the **label** option is omitted. To read an ANSI labelled tape from a single device with tape file name "t," expiration date 9/9/96 and density of 1600 BPI :

```
from (16 name t volid 000069 expdate 09/09/96 den 1600)
```

If you specify the **label** option without a label type, ANSI labelling is assumed. To read an ANSI labelled tape from a single device with tape file name "t," volume ID 00069 and an expiration date of 9/9/96:

```
from (16 name t label volid 000069 expdate 09/09/92)
```

Used with

copy	listdir	listinfo
reload	restore	validate

ignlabel

Omits the RoadRunner tape label check.

Syntax `ignlabel`

Comments If you do not use RoadRunner labels to provide overwrite protection, you can specify the **ignlabel** keyword for backup operations. When you specify **ignlabel**, RoadRunner does not check for a RoadRunner label prior to writing tape. It still checks for ANSI labels.

WARNING

If you specify **ignlabel**, the RoadRunner label *is not checked* and no overwrite protection is provided. Therefore it is possible to overwrite a store tape that has a RoadRunner label with an unexpired date. Also, unless you use ANSI labels, RoadRunner does not check to ensure that you do not overwrite a previous reel from the same store set.

For example, assume you have filled the first two reels of your backup and are ready to mount the third. If you were to accidentally mount one of the two reels just written, RoadRunner would normally warn you of your mistake and prompt you to mount another reel, as shown in the example below:

```
The currently mounted reel is incorrect
Please mount next volume (reel # 3) on LDEV 7
```

However, if you specify **ignlabel**, RoadRunner does not perform this check, which makes it possible to overwrite a previously-written reel of the current store set.

This keyword can only be specified globally, once in a command.

Example The following command begins a store without checking the mounted volume for a RoadRunner label.

```
select @.@.dev store to 17 ignlabel
```

Used with

`copy`	`full`	`incremental`	`interim`
`listdir`	`partial`	`reload`	`restore`
`store`	`validate`		

include

Includes the contents of a text file in a RoadRunner command at execution time.

Syntax **include** *filename*

Comments The **include** keyword allows you to save parts of a command in a file and splice it in during execution. The resulting command behaves as if the text from the include file were entered at the location of the **include** keyword. Include files can be nested up to 10 levels deep.

include can be specified at any point in a command, as many times as you like. The text in the file is read in as soon as **include** is encountered.

Example In the example below, an include file is written and then used in a command. This include file, repfmt, contains commands that purge the existing *output* file if one exists, writes a file equation for the output destination, and defines the report format. This file could be used to manage output from a variety of commands. In this example, it is used with **listdir**.

```
:run rr.pub.tym
<01>mpe "purge output"
<02>mpe "file output; rec=-80, , f, ascii; disc=20000"
<03>report (short, filecode option 4up to *output)
<04>/
<rr>keep repfmt
<rr>new
<01>include repfmt listdir from 17
<02>/go
```

Include files also provide a function similar to BackPack or STORE USE files, as they can contain the entire text of a command. For example, assume the following include file named nojunk:

```
mpe file purgelst,old;acc=append
select (@.pub.sys where credate > lastfull and creator <> manager
purge after each "echo \filename >> purgelst")
select (@.pub.sys)
select (@.@.@ excluding @.pub.sys)
full to (tape label ansi name f1) and (tape label ansi name f2)
```

You could execute this file as follows:

```
run rr.pub.tym;info="include=nojunk"
```

Used with Can be used as a replacement for any set of commands or keywords, or by itself if the included file contains the full set of RoadRunner commands.

index

Used during restore and copy operations to select a particular backup to read from in an appended set. See "Appending Multiple Backups on DAT Cartridges" on page 2-13.

Syntax `index [nn |name backupname |ALL]`

where *nn* is the index number of the backup (from 1 to 512) and name is any string up to 35 characters in length that identifies the backup. The **index all** option is only allowed with the **listdir** and **listinfo** commands. This option lists the directory or lists the information for all appended backups on one media volume.

Comments **index** can be specified only once in a command.

Example To restore the fourth backup appended to a cartridge set:

```
restore from DAT index 4
```

To validate the backup named partial appended to a cartridge set:

```
validate from 7 index name "partial"
```

To validate the second backup appended to a cartridge set and report the total number of restorable files as well as the file names of unrestorable files and the reasons they can't be restored:

```
validate from DAT index 2 report (with nofiles, header, command)
```

To copy from an appended backup named "tuesday" on device 7 to an appended backup named "archive94" on device 9, overwriting the existing appended backup named "archive9".

```
copy from 7 index name "tuesday"
to 9 append name "archive9"
backup name "archive94"
```

Used with

copy	listdir	reload
restore	validate	listinfo

interleave

Reads data from a specified number of files concurrently.

Syntax `interleave` *n*

where *n* is an integer from 1 to 8.

Comments If **interleave** is not specified, RoadRunner automatically sets the interleave level to 4. The maximum number of files RoadRunner can interleave is 8.

Specifying a high interleave value on backup causes data to be backed up faster, but restoring the data may take a little longer.

interleave can be specified only once within a command.

Example In the following command, RoadRunner backs up files in the dallas group with an interleave value of six.

```
select @.@.dallas interleave 6 store to 17
```

Used with

`full` `incremental` `interim` `partial` `store`

keep/ nokeep/ keepnew

keep specifies that if a file on tape has the same name as a file on disc, the file on tape is not restored. **nokeep** specifies that the file on disc is overwritten. **keepnew** specifies that the newer of the two files is retained.

Syntax

```
keep
nokeep
keepnew
```

Comments

nokeep is the default for the **restore** command. **keep** is the default for **reload**.

keep, **nokeep** and **keepnew** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

The following example overrides the default **keep** option for the files in the first **select** statement. The files specified in this **select** statement overwrite existing versions on disc.

```
select (command.pub.sys, @.net.sys, nmconfig.pub.sys nokeep)
select @.@.@
reload from 7
```

Used with

```
reload  restore
```

local/nolocal

local places restored files in your logon group and account and changes the creator name of each file to your user name. **nolocal** disables this function for a selected fileset and is the default.

Syntax

```
local
nolocal
```

Comments

The **local** keyword is useful when restoring files that were stored on a system with a different directory structure. **change** and **create** can also be used for this purpose.

local can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

The following example restores files from group1.acct1 into the logon group and account.

```
select @.group1.acct1
restore from 7 local
```

Used with

reload restore

lock/nolock

The **lock** keyword specifies that write access to files is denied during backup. The **nolock** keyword allows write access to files during backup. The default is **lock**.

Syntax

```
lock
nolock
```

Comments

When you specify **nolock**, RoadRunner does not set the store bits of files selected to be backed up. Normally, store bits are set to prevent files from being written to by other processes during the store.

When **nolock** is used, **select** statements with overlapping fileset specifications cause those filesets to be stored as many times as they are selected.

If **dbstore** and **nolock** are both specified, **dbstore** overrides **nolock** for databases only, preventing write access to database files.

WARNING

If you use **nolock**, you must ensure that no data is added to or deleted from any file on your system during the store.

Because RoadRunner does not set the store bits checked by MPE iX before allowing users to access files, nothing prevents a user from running a program that opens data files during a RoadRunner store operation running in **nolock** mode. Files opened with WRITE or EXCLUSIVE access at file copy time are not stored and are shown in SYSLIST as NOT STORED for one of these reasons:

```
File was open for write access
or
File was open exclusively (NOLOCK in effect)
```

Renaming or purging files must also be avoided to prevent errors.

lock and **nolock** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Online Module

See also **online** (online module only) and **unlock**. If **online** is specified, neither **lock** nor **nolock** can be used.

Example

In the following example, a full backup is performed in a stand alone situation, so **nolock** can be used. To prevent users in the dallas account from modifying their files during the backup, **lock** is used on their account. The dallas account is excluded from the first **select** to prevent its being backed up twice.

```
select (@.@.@ nolock excluding @.@.dallas)
select (@.@.dallas lock) <-- The lock keyword is not really required
store to 17                  since it is the default.
```

Used with

full incremental interim partial store

maxerrs

Specifies the maximum number of soft errors allowed before RoadRunner rejects the current tape and initiates error recovery procedures.

Syntax

maxerrs [=] *nn*

where *nn* is the number of soft errors.

Comments

A soft or recoverable tape error is one in which data cannot be written to tape on the first attempt, but then is written successfully on a subsequent attempt. By tracking these errors and reporting when more than ten occur in the course of a particular reel, RoadRunner alerts you that either the tape in use or the tape drive itself is not delivering acceptable performance.

With the **maxerrs** keyword, you can specify the number of soft errors after which you want RoadRunner to invoke the error recovery procedures it uses when an unrecoverable tape error occurs. This means that RoadRunner rejects the current tape, requests that you mount a new one, and starts the new reel with the first block of data written on the rejected reel.

Whether or not **maxerrs** is specified, RoadRunner always reports the number of soft errors that exceed 10. If **maxerrs** is not specified and ten or more soft errors occur, RoadRunner prints a warning, but does not automatically initiate recovery procedures.

maxerrs can be specified only once in a command.

Example

In the command below, **maxerrs** causes error recovery procedures to be initiated if five or more soft errors occur.

```
full to 17 maxerrs=5
```

Used with

copy	full	incremental	interim
partial	store		

mpe

Executes any programmatically executable MPE iX command, command file, or UDC before execution of a RoadRunner command.

Syntax **mpe** *command*

Comments The **mpe** keyword embeds MPE iX commands that are executed before the RoadRunner operation. This keyword allows you to include file equations in RoadRunner commands to provide jobstream-like capabilities.

To use double quote marks in an MPE iX command, place a backslash before each double quote. The backslash also designates nonprintable characters, such as carriage returns, etc.

`\"`	Designates a quote
`\0`*nnn*	Designates a nonprintable ascii character of value octal *nnn*
`\`*nnn*	Designates a nonprintable ascii character of value decimal *nnn*
`\0x`*nnn*	Designates a nonprintable ascii character of value hex *nnn*
`\`	When not followed by a digit or a quote, a backslash is interpreted literally.

mpe can be specified at any point in a command and as many times as you like. RoadRunner MPE commands are included in the command buffer of RoadRunner. These commands will be performed before the rest of the command is processed. If you want **mpe** commands executed immediately by the command interpreter, do not precede with the **mpe** keyword. Instead use the exclamation point (!) followed by the command string, or the colon (:), if interactive.

Example The following example purges any file named listfile and sets up a file equation for **report** to use:

```
mpe "purge listfile"
mpe "file listfile; rec=-80, , f, ascii"
mpe "file t; dev=tape"
listdir from *t report (to *listfile)
```

This example illustrates the use of quotes within the command:

```
mpe "run myprog; info=\"initial\"initial\"
```

Used with

copy	full	incremental	interim	store
listdir	partial	reload	restore	validate

nopriv

Sets the file code which RoadRunner uses for creating VT (virtual tape) backups to a positive (nonprivileged) value.

Syntax `nopriv`

Comments RoadRunner normally uses a negative file code (-21074) when storing to VT files. This designates that the VT files are "privileged files" to the MPE file system and special restrictions are placed on applications accessing them. These restrictions provide important security and reliability elements for VT files used as normal backups.

RoadRunner is being used more and more as a powerful file management tool. In this mode, the convenience of being able to handle a nonprivleged VT file using traditional tools has become more important. To allow this behavior, the **nopriv** keyword can be specified when VT files are created to cause the file code to be set to a (positive) 21074. These files can then be purged, copied, or moved to other systems using conventional methods for handling binary data files.

Note

However, any data stored in a VT file which is not privileged can be read by any user on the system who has normal MPE read access to the file. This would include any directory information stored to the VT file such as user, group, and account passwords.

When reading from a VT file, the file code can be either +21074 or -21074 and RoadRunner will attempt to process the file normally. However, files with any other file codes will be rejected as invalid VT files unless the **recover media** option is enabled. If **recover media** is specified and the VT file has a nonstandard file code, a warning will be generated and RoadRunner will attempt to process the file as a VT file.

The **nopriv** keyword can only be specified globally, once in a command.

Example To generate a nonprivileged VT file or set of files:

```
select @.pub
store to (disc name dbvt.pub.acct) nopriv
```

Used with

copy	full	incremental	interim
partial	store		

noroadrest

Disables placement of a copy of RoadRunner's restore module in MPE iX STORE format at the start of a tape.

Syntax

```
noroadrest
```

Comments

Unless you specify ANSI tape labelling, RoadRunner automatically includes a copy of its restore module in MPE iX STORE format at the beginning of the first volume of a backup to tape. This is useful for disaster recovery or distribution of files to other sites: you can restore files stored with RoadRunner on any MPE iX system, even if it does not have a copy of RoadRunner.

If you know that you won't need to restore on a system without RoadRunner, you can reduce elapsed time and tape space with **noroadrest**.

If an ANSI labeled tape is specified, the ROADREST program is not placed at the start of the first reel, so the **noroadrest** keyword is not needed.

noroadrest can be specified only once in a command.

Example

When storing a small fileset for your own use, including a copy of RoadRunner in the backup is superfluous.

```
select myfile.mygroup.myacct store to tape noroadrest
```

Used with

```
full   incremental   interim   partial
store  copy
```

olddate/newdate

When **olddate** is specified, restored files retain their original creation dates. When **newdate** is specified, creation dates are changed to the date of the restore. **newdate** is the default.

Syntax

```
olddate
newdate
```

Comments

To ensure that the access dates and modification dates of the restored files remain unchanged, always add the **olddate** keyword to the **restore** command. If **olddate** is not specified, RoadRunner changes these dates to the date of the restore, just as MPE iX :RESTORE does. For the **reload** command, **olddate** is the default.

olddate and **newdate** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

In the following command, **olddate** is specified to keep the access and modification dates of the files being restored.

```
select @.pub.sys restore from 17 olddate
```

Used with

reload restore

Online Module

online

Initiates backup concurrent with user access to files. See "Online Backup" on page 3-16.

Syntax

```
online
```

Comments

For sites that require minimum downtime, RoadRunner's online backup approach provides read and write access to files throughout the store process with no perceptible degradation of system response.

At the start of an online backup, users must close their files briefly so RoadRunner can establish a "synchronization point." During this time RoadRunner reads fileset specifications, builds its file directory, and flags the files to be backed up. Any files open for write access during the synchronization period are not stored.

online can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

To initiate an online backup of all files, have users close their files, then enter the following command. The **after directory** option is used to notify users when the synchronization period is over.

```
full to tape online
after directory "tell @.@ All files are ready for access"
```

Used with

```
full   incremental   interim   partial
store
```

onvs/notonvs

When **onvs** is specified, RoadRunner restricts file and directory selection to the volume sets specified. **notonvs** excludes the files and directories on specified volume sets.

Syntax

```
onvs volsetlist
notonvs volsetlist
```

where *volsetlist* is one or more valid MPE iX volume set names consisting of 32 or less alphanumeric characters and the underscore. The elements of the list are separated by commas. It can include the wildcards listed on page 2-23.

Comments

The **onvs** keyword can be added to backup commands to specify volume sets to be copied to the backup media. Only files on the volume sets listed are stored.

The *volsetlist* is applied as a restriction to the *filesetlist*. If no files in the fileset specified fall within the volume sets listed, no files are stored.

To store directory information for the volume sets, add the **directory** keyword. Only the directory structures for the specified volume sets are backed up.

You can store only the directory structures with no data files by omitting the **select** keyword.

onvs and **notonvs** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

onvs and **notonvs** are mutually exclusive. You cannot apply them to the same fileset, or use them globally within the same command.

Examples

The following command backs up files on the nonsystem volume set called prod_volume_set and the system volume set, along with their directory structure.

```
select @.@.@ onvs prod_volume_set, mpexl_system_volume_set
store to 17 directory
```

The following command backs up all volume sets except for gl_volume_set, along with their directory structure. (Use this if a particular volume set is unmounted at the time of the backup due to disc failure, etc.)

```
select @.@.@ notonvs gl_volume_set
store to 17 directory
```

The following command backs up all volume sets except for gl@, along with their directory structure.

```
select @.@.@ notonvs gl@
store to 7 directory
```

Used with

full	incremental	interim	partial
scan	store		

priority

Sets the MPE/iX priority queue for a RoadRunner process.

Syntax

priority *queue* **[,nolinear]**

where *queue* is:

bs | cs | ds | es

where **nolinear** forces RoadRunner processes to be scheduled by the dispatcher.

Comments

The processing queues on your system determine how much resource is devoted to the operations that run in those queues. BS is the highest priority queue; ES is the lowest. If **priority** is not used, RoadRunner processes run in the logon priority queue of the user or job stream. Within the selected queue, RoadRunner uses the linear queue as opposed to the circular queue, establishing fixed priorities for each process to provide optimum performance. Note that the MPE SHOWPROC command and various third-party performance utilities show processes executing in the linear queue as being in the BS queue, although their actual priority values fall within the limits set on your system for whatever queue they are running in. So, for example, even if you are running RoadRunner in the CS queue, SHOWPROC will indicate that RoadRunner is in the BS queue.

If performance is not an issue and you want RoadRunner to share CPU service equally with other processes executing in the same priority queue, you can add the **nolinear** option to force RoadRunner processes to run in the circular queue. This typically degrades RoadRunner performance, even if no other processes are executing.

priority can be specified only once in a command.

Example

In the following command, the *queue* is set to the highest allowed priority.

```
full directory dbstore to 17 priority bs
```

Used with

copy	full	incremental	interim
listdir	partial	reload	restore
scan	store	validate	

purge/nopurge

When **purge** is specified, RoadRunner deletes disc files after they are stored to tape. **nopurge** disables this function for a selected fileset and is the default.

Syntax

```
purge
nopurge
```

Comments

Before deleting a file, RoadRunner performs two security tests.

- It verifies that the file has been successfully stored to tape. If for any reason it was not, it is not deleted.
- It checks to ensure that you have write access to the file. If you do not, the file is not deleted. Users with SM capability have write access to all files on the system. Users with AM capability have write access to files in their account.

purge and **nopurge** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Examples

To store the old account to tape and purge all files in the account except those in the pub group:

```
select @.@.old store to (17 name backup) purge
select(@.pub.old nopurge)  ← Overrides purge locally
```

To back up the entire system and delete files in the temp account:

```
select (@.@.temp purge)
select (@.@.@ excluding @.@.temp)
full to (17 name backup)
```

Used with

```
full  incremental  interim  partial  store
```

purgebefore/ purgeafter

When **purgebefore** is specified for a **restore** operation, files to be replaced are purged on a file by file basis before each one is replaced. When **purgeafter** is specified, the disc file is not purged until the backup file is restored. The default is **purgeafter** *unless* RoadRunner runs out of disc space while restoring a file. At that point it operates like **purgebefore**: the old file is purged and RoadRunner attempts to spread the new file across volume class `DISC` within the current volume set. Specifying **purgeafter** cancels this default response.

Syntax

```
purgebefore
purgeafter
```

Comments

purgebefore increases the free disc space prior to restore operations by clearing the disc of the file to be replaced. When restoring a large file in a limited disc space situation, it may be useful to purge the copy on disc before restoring the backup copy to ensure there is sufficient disc space for restore.

purgeafter guarantees that a good copy of the file is available if the backup copy proves unrecoverable.

WARNING

If **purgebefore** is specified and the backup copy of a file is unrecoverable due to a media error, the file is lost. You will need to restore the file from another backup.

purgebefore and **purgeafter** can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Example

In the command below, use of disc space and security of data are optimized by using **purgeafter** on small files, and **purgebefore** on large ones:

```
select (@.@.@ where space > 100000 purgebefore)
select (@.@.@ where space <= 100000 purgeafter)
restore from 17
```

Used with

`restore` `reload`

recover

Attempts to restore files from a tape set which is either damaged or has missing volumes. See "Error Recovery" on page 3-28.

Syntax `recover [media | files | all]`

Note

When using the **recover** keyword, it is recommended to use the **concurrent** keyword (see page 5-18), so the report will show the file being restored.

Comments There are three areas on a RoadRunner tape: the header that identifies the tape as a RoadRunner tape; the directory of files; and the data. Media errors can occur in any of these areas and RoadRunner provides recovery options for all of them. RoadRunner also provides recovery if you are missing reels from a tape set.

recover media is used to recover files if errors occur in the header or the directory. This option causes RoadRunner to continue to attempt to restore files rather than quit automatically if an out-of-sequence reel or a reel with an unrecognizable label or directory is mounted.

recover files restores all possible data when errors occur in the data portion of the file.

recover all invokes both types of recovery. This is the default.

Example The following command restores a tape on logical device 7. Specifying **recover media** causes RoadRunner to ask you whether to continue rather than quit automatically if an out-of-sequence reel or a reel with an unrecognizable label or directory is mounted.

```
restore from 7 select @.@.dallas recover media
```

Used with

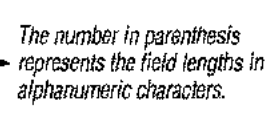
`reload restore validate`

report

Specifies format, content and destination of listings for RoadRunner operations.

Syntax `report [([fields] [with options] [to devs] [format formats])]`

where *fields* is one or more of the following field options, separated by commas:



`accountname**`	Name of account. (8)
`acctime`	Time of last access to file. (7)
`blockfactor`	*Blocking factor of file. (3)
`blocksize`	Size of block.(9)
`compression`	Compression level specified, percentage compression achieved and throughput in megabytes per second achieved for this file. (13)
`creator`	File creator. (8)
`credate`	Date of file's creation. (8)
`cretime`	Time of file's creation. (7)
`dates`	Creation, modification and access dates. (26)
`eof*`	Number of records in file. (9)
`extents*`	Number of extents in file. (5)
`filecode**`	Numeric or mnemonic MPE/iX file codes for file. (5)
`filename**`	Name of file. (8)
`fullname`	Filename.groupname.accountname. (26)
`groupname**`	Name of group. (8)
`limit`	File limit in records. (9)
`lockword`	File lockword. Not accessible unless you have SM or AM capabilities, or you are the file creator. (8)
`long`	Includes: *filename, groupname, accountname, volinfo, sectors, filecode, media, recsize, rectype, eof, blockfactor, extents.* (131)
`media**`	Volume number on which the file was stored. If parallel devices were used, includes set number and reel number. (6)
`moddate`	Modification date of file. (8)
`modtime`	Modification time of file. (7)
`pathname`	Fully qualified POSIX pathname for file. (26)
`ospfname`	Original spool file name at time of backup. (8)
`recsize*`	Size of record, in bytes. (6)
`rectype*`	Type of record, for example, fixed, variable, ASCII, binary, etc. Variable is the default. (4)
`sectors**`	Number of uncompressed sectors in a file. (7)
`security`	Security matrix for file. (75)
`short`	Includes: *filename, groupname, accountname, volinfo, sectors, filecode, media.* (79)

`volinfo**`	Volume restrictions specified in the file label. (31)
`volname`	Volume ID of a labelled tape. (6)

where *options* is one or more of the following, separated by commas:

`command`	The RoadRunner command that initiated the operation.
`header`	See listing on page 5-58.
`nofiles`	Omits information about files successfully processed.
`norejects`	Information for only those files RoadRunner was able to process; files rejected for any reason are not listed.
`volnames`	The volume IDs of labeled tapes used.

where *devs* is one or more of the following, separated by commas:

`dev#`	The specified logical device number.
`devclass`	The specified device class.
`*file`	A file specified in the back-referenced file equation (usually a disc file).
`offline`	The line printer (device class `LP`, designator `OFFLINE`).
`stdlist`	The listing device for your session.

where *formats* is one or more of the following, separated by commas:

`A4`	A4 page size for printers that support this option.
`2up`	2 reduced pages of data on each sheet. For HP Laser Jet compatible printers only.
`4up`	4 reduced pages of data per sheet. For HP Laser Jet compatible printers only.
`duplex`	On front and back of page. For HP Laser Jet compatible printers only.
`environment filename`	Uses the specified environment file. No other format options except **lines** and **width** are permitted.
`landscape`	Pages oriented on long edge. For HP Laser Jet compatible printers only.
`lines nn`	*nn* lines per page (Used only to override your printer's built-in setting). Adds a banner and number to the top of each page.
`portrait`	Pages oriented on short edge. For HP Laser Jet compatible printers only.
`width nn`	*nn* characters per line (Used only to override your printer's built-in setting).

* - Included in the long field option.
** - Included in both the long and short field options.

Comments To produce one or more listings either in the default format, or with the fields, destinations, and page layouts you specify, use the **report** keyword as many times as needed within a command. If **report** is not specified, RoadRunner still provides a list of files excluded from the operation and the reason each file was excluded. The exception to

this is the **listdir** command, which has its own default listing. See the **listdir** command on page 4-9.

If **report** is specified with no options, RoadRunner provides information about the files stored in addition to the list of files excluded. What information listed about the files stored depends on whether RoadRunner is running in batch mode or interactive mode. In batch mode, the default is **long**. In interactive mode, the default is **short**. See the description of these options on page 5-55.

If you specify field options, RoadRunner includes only the specified fields listed in the order you specified them. However, if your list does not include the *filename* option, the file name, group name, and account name for each file are printed in the first three columns of the report. If POSIX files are selected, these files get file name, group name, and path name.

If no devices are specified, the report is sent to `SYSLIST`, which defaults to `$STDLIST` unless overridden with a file equation. In batch mode, `$STDLIST` is the output device designated for that job (usually the system line printer). In interactive mode, it is the terminal from which the operation was initiated. If a file name is specified and that file already exists, and if you have not opened it with append access (`ACC=APPEND`) in your file equation, RoadRunner builds an alternate file for the report output called RRhhmmss, where hhmmss is the time the file was created.

If you are reporting `to` a file (for example, report to *x) and the file referred to (*x) already exists, your report is saved as RRhhmmss, where hhmmss is the time of day that the report is generated.

report cannot be specified locally. Though it can be used more than once in a command, its application is always global.

If any POSIX filesets are to be backed up, this pathname replaces the `FILENAME`, `GROUP`, and `ACCOUNT` options in the **long** and **short** fields so that if any HFS files are selected, the file's pathname is printed in the last column of the report.

If *pathname* is specified for files in MPE namespace, RoadRunner maps the MPE filename to a POSIX filename. For example, the standard MPE `FILE.GROUP.ACCOUNT` filename is displayed as the hierarchical file structure: `/ACCOUNT/GROUP/FILE`.

If the *filename* option is specified for a file that is not a descendant of an MPE account or group, the POSIX file name (the last element of the path name) is stripped off and used. Since POSIX file names can be longer than MPE file names, the first eight characters are printed.

If the `ACCOUNT` or `GROUP` attributes are specified and the file is not in MPE namespace, `root` is printed for those attributes.

Due to the potential length of POSIX path names, some provisions have been made for formatting the *pathname* attribute.

The default width of the *pathname* field is 26 characters (the same as the `FILE`, `GROUP`, and `ACCOUNT` attributes combined). If the path name is too large to fit on one line, the path name wraps automatically to the next.

For information on listing formats and further examples of the **report** keyword, see "Listing Formats" on page 3-21.

Examples

The following example generates a default report. The **long** format is used if the output file is 132 characters or more wide; otherwise, the **short** format is used. The report is sent to the formal file designator `SYSLIST`, which defaults to `$STDLIST`, unless overridden by a file equation:

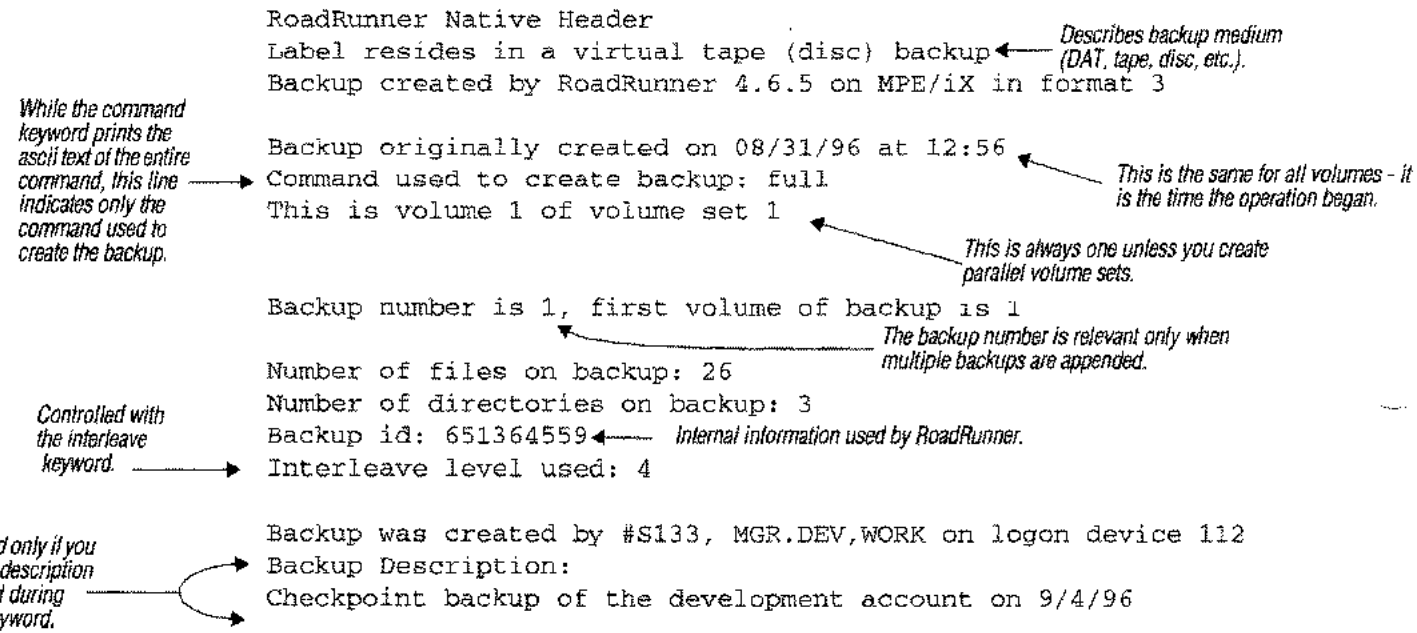
```
full to (tape name t)
report
```

The following example sends a 2-up listing in the short format to a LaserJet and a list of file names to device #65:

```
mpe "file ljet;dev=25"
full to (tape name t)
report (short to *ljet format 2up)
report (filename to 65)
```

Backup Header

A sample backup header, produced with the **report (with header)** option, follows. RoadRunner produces one of these for each volume, or for each backup in the case of appended backups.



A sample short listing of POSIX files is provided:

Volume Restrictions	Sectors	Code	Media	Pathname
DISC	:C	256	1	/usr/f2
DISC	:C	256	1	/usr/files/f2
DISC	:C	256	1	/usr/files/f3
DISC	:C	256	1	/usr/files/f4
DISC	:C	256	1	/usr/files/f5
DISC	:C	256	1	/usr/test/file
DISC	:C	256	1	/usr/usr2/f1
DISC	:C	256	1	/usr/usr2/f2
DISC	:C	256	1	/usr/usr2/usr3/f1
DISC	:C	256	1	/usr/usr2/usr3/f2
DISC	:C	256	1	/usr/usr2/usr3/usr4/f1
DISC	:C	256	1	/usr/usr2/usr3/usr4/f2
12 files were listed				

Used with

copy	full	incremental	interim
listdir	listinfo	partial	reload
restore	scan	store	validate

select

Specifies files to be processed.

Syntax

```
select fileset [,fileset]...
select (fileset [,fileset]... [keyword [keyword]...])
```

where *fileset* is one of the following:

filename The file or files to be processed; may contain wildcards and follows standard MPE-style or HFS-style file naming conventions.

^indirectfile

A text file listing fully-qualified file names (wildcards permitted) delimited by blanks or carriage returns. Comments, delimited by braces ({ and }), are allowed anywhere in the file and can span multiple lines.

where **keyword** is one of the following:

RoadRunner supports the wildcard characters listed on page 2-23.

Note

Under POSIX, all directories, including MPE (non-hierarchical) directories are **subsets** of the HFS root, indicated by a slash (/). Use a slash (/) to select all files. If you use @.@.@ to select all files, a message will be displayed indicating that this entry has been interpreted as a slash (/).

RoadRunner recognizes a fileset to be HFS-style when the first character is either . or /. To continue a path name that does not fit on one line, place a backslash (\) at the end of the line and continue the path name on the next line.

If the last component in the selection path matches a file name only, and the path name ends with a slash (or **tree** is specified), nothing is selected. If the last component in the

selection path matches a file name only, and the path name does not end with a slash (or **notree** is specified), RoadRunner selects the matching file only.

To exclude files from the fileset, use the **excluding** keyword. To qualify files based on virtually any file attribute, including modification/access date, use the **where** keyword. See Examples below.

Though indirectfiles are supported for the convenience of users migrating from other backup software, RoadRunner's include file capability provides greater flexibility and better performance. See the discussion on page A-5. The **include** keyword is covered on page 5-38.

The effect of any keyword listed on the preceding page can be **local** (restricted to the files indicated in one **select** subset) or **global** (applying to all files in the command, both inside and outside parentheses). To apply a keyword locally, enclose it in parentheses following the **select** keyword. To apply a keyword globally, specify it outside parentheses used to enclose **select** subsets. Local specifications override global ones for the filesets to which they apply. For example:

```
select (@.pub.myacct nopurge)
select @.@.myacct purge
```

In this case, for the @.pub.myacct fileset, the local specification of **nopurge** overrides the global **purge**.

If **select** is not specified, the default fileset depends on your MPE/iX capabilities:

	Limits	Defaults - MPE 4.0 or earlier	Defaults - MPE 4.5 or later
SM or OP	Can store all files	`@.@.@`	`/`
AM	Can store all files in your account or that you can read	`@.@`	`/account/`
all other	Can only store files you can read	`@`	`./`

When you use several **select** statements, RoadRunner stores files in the order the **select** statements appear. If you select a file more than once for a store operation, theoretically the file is backed up as many times as specified, each time applying any local options indicated. In fact, unless you use the **nolock** keyword, the second and subsequent attempts to select the file will fail due to the problem "file being stored by another store process" and end up in the "files not stored" list.

Spoolfiles that are printed while their storebit is on are not purged by MPE after printing. To ensure that these files are purged, exclude them from your **select** statement (page 5-33), or use the **nolock** keyword (page 5-43) to keep their storebits from being set.

On **restore**, **reload**, **validate**, and **copy**, the first time that the file is selected is the only one that has any effect. The local options applied to it in that select are used.

Examples

The following command selects a number of files. The first selects @.pub.parts, @.pub.gl, and @.pub.dev, the second selects all files that were created after 1/1/96 and files in the

.sys account that were created in the last 30 days, and the last selects all files in the infosys account that are not image databases.

```
select @.pub.parts, @.pub.gl, @.pub.dev
select (@.@.sys where credate > today-30)
select (@.@.infosys excluding (@.@.@ where isimage))
where credate > 1/1/96
store to *t;
```

To select all files in all accounts having names that begin with the letters A through R, the following command would be used.

```
@.@.[A-R]@
```

In the following example, an indirectfile is used to select a predefined list of files. The files selected in the indirectfile are then stored to virtual tape:

```
mpe "file vt;dev=disc"
select ^myfiles store to *vt
```

myfiles might look like this:

```
@.source.dev
udc03.udc.sys
prodlist.pub.dev
spec@.@.specs
@.fldwrk.tom
```

The following example uses POSIX file selections:

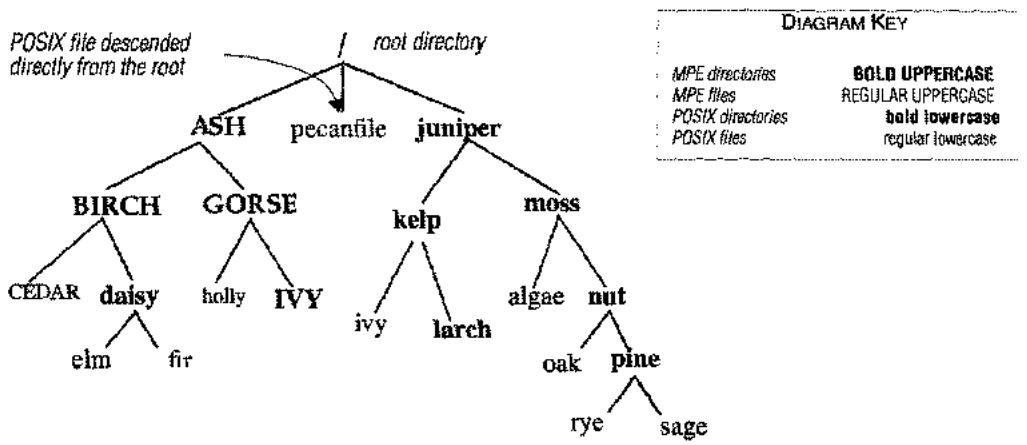
```
select/usr
select/users/%/.profile
```

Some examples of file selections using wildcard in a HFS environment follow. All directories in the examples refer to the directory structure represented above. As in the diagram on page 1 of this update, upper case letters represent MPE structures, while lower case letters represent POSIX structures.

In the following examples, **usename** syntax is used:

Using...	Chooses...
/	All the directories and their contents under the root.
/%/@	All the directories and files under the root.
/@/	The contents of directories and the directories under the root, except for the pecanfile (a file located in the root directory).
/@	All files in the root directory (in this case, the pecanfile).
/ASH/	The files in directory ASH, and all subdirectories and their contents that are descended from ASH.
/%/[Cif]@	All files below the root that begin with C, i or f. Using the example diagram, this would include the file CEDAR in MPE namespace and the POSIX fir and ivy files.

Using...	Chooses...
/%/[GI]@/	All files in directories that begin with G or I. Using the example diagram, this expression chooses the files in the IVY directory and holly file in the GORSE directory.
/%/[a-f]@	All files beginning with any letter in the range from lowercase a to lowercase e (algae, elm, and fir from the sample diagram).
/juniper/kelp/larch/	All files in the larch directory.



The following examples illustrate the use of wildcard characters but do **not** refer to the sample directory.

Using...	Selects...
/%/@.c	All files on the system that have names ending in .c
/%/source/@.h	All files with names ending in .h in any subdirectory named source.
/%/austin/%/dev/%/@tom	All files that have austin and dev somewhere on their path and have names ending with the letters tom.
/%/tom@	Any files that begin with the letters tom along with all files in any directories that begin with the letters tom.
/%/tom@/	All files in any directory that begin with the letters tom along with all files in all directories below them.
/austin/	All files in the austin directory and all directories below it.
/austin	If austin is a file, it is selected, if it is a directory, all files in it are selected, but none of its dependent directories are selected.

Using...	Selects...
`./`	The current working directory and its descendants.

The following examples illustrate the use of wildcard characters in MPE commands.

Using...	Selects...
`@.@.@`	All the directories and their contents under the root.
`@.@.ACCT`	Only files in MPE namespace under ACCT.
`@.@.[?]@`	All files in MPE namespace.

Used with

```
copy     full     incremental   interim
listdir  partial  reload        restore
scan     store    validate
```

single

Validates or restores files from just one reel of a tape set.

Syntax **`single`**

Comments If a large number of soft errors occur on one of the reels of a tape set, you may want to verify the restorability of just that reel. To do so, specify the **single** keyword with the **validate** command.

If some of the reels of a tape set have been lost or damaged, you can restore files from just one reel by adding the **single** keyword to your **restore** or **reload** command. This is similar to using the **recover media** option.

If there are files on the reel you restore or validate that span to the next volume, RoadRunner prints messages like those shown below. The number of incomplete files is usually equal to the interleave level in effect (the default is 4).

```
RoadRunner WARNING 17: Some files were not complete (SINGLE in effect)
RoadRunner WARNING 18: Data is missing for file TEST.MYGRP.MYACCT

FILENAME.GROUP   .ACCOUNT NOT STORED BECAUSE
TEST   .MYGRP  .MYACCT  Data for file was incomplete (SINGLE or RECOVER in
effect)
```

Example In the command below, a single reel is validated.

```
validate from tape single
```

The following command restores all files from a single reel of a tape set. Even if this is not the first reel, you are not prompted as usual for the first reel.

```
restore from tape select @.@.@ single
```

Used with

`reload` `restore` `validate`

splitvs

This keyword is used to support the Hewlett-Packard Mirrored Disk/iX facility.

Serves as a selection criteria similar to the **onvs** keyword except the **splitvs** keyword tells RoadRunner to process only the "backup files" that have been specified.

Syntax

splitvs *volsetlist*

where *volsetlist* is one or more valid MPE iX volume set names consisting of 32 or less alphanumeric characters and the underscore. The elements of the list are separated by commas. It can include the wildcards listed on page 2-23.

Comments

The **splitvs** keyword affects which files are selected from any given volume set as follows:

1. The **splitvs** keyword serves as a selection criteria, in the same manner as the **onvs** keyword. This means that only files which reside on volume sets specified in the onvs or **splitvs** keywords will be selected. Files which are on other volume sets will not be selected.
2. Unlike the **onvs** keyword, the **splitvs** keyword, *without wildcards*, informs RoadRunner that only the "backup copies" of files on the specified split volume sets are to be selected. When wildcards are not used and the volume set is not split, the files on that volume set will not be selected. An error message will appear if the volume set is not split.
3. When using **splitvs** *with wildcards*, even if the volume set is not split, files will be selected and no warning message will be displayed. If the volume set is split, files on the volume set will be selected from the backup copy instead of the user copy.
4. The **splitvs** and **onvs** keywords may be used together. If a volume set qualifies for both the **onvs** and **splitvs** keywords, the **onvs** keyword is ignored.

If any groups are purged from the system during the backup, information (files or directories) from those groups will not be backed up. This is a restriction of the MPEiX operating system.

Example

A mirrored volume set must be "split" to perform an "online" backup which allows a backup to be done while users continue to access those same files on the "user" copy of the volume set. This online method is not related to RoadRunner's ability to perform backups using the **online** keyword. This can be accomplished as follows:

1. Make sure that all users of split volume sets are logged out. All files *must* be closed at this point.

Note

You might want to use the **vsuser** command before the **vsclose** command to check for users with open files on the mirrored volumeset.

2. Use the MPE VSCLOSE command to "split" each volume set:

   ```
   :VSCLOSE setname; SPLIT
   ```

   This step makes both sets of drives unavailable.

3. Re-open the volume set with the MPE :VSOPEN command:

   ```
   :VSOPEN setname
   ```

This step makes the user set of the mirrored drives available for use again.

4. Do your backup.

   Use the **splitvs** keyword to backup the files on the backup copy of the mirror disk pair.

   ```
   SPLITVS @ FULL
   ```

   In this example, backup all volume sets on the system, however, any volume sets that are split will select the files from the backup copy. Files on volume sets that are not split will be selected.

5. Finally, you must rejoin and resynchronize the mirrored sets:

   Since data has probably changed on the active copy of the mirrored disk set while the backup was running, the two sets must now be re-synchronized. Use the MPE/iX VOLUTIL system:

   ```
   :VOLUTIL
   >JOINMIRRSET setname SOURCE=USER
   >EXIT
   ```

Used with

```
full      interim incremental   partial
scan      store
```

spoolpri

Specifies output priority for restored spool files.

Syntax

spoolpri *nn*

where *nn* specifies the output priority of restored spool files; 0 - 13

Comments

By default, RoadRunner restores spool files with the same output priority they had at the time they were stored. (This matches the defaults of HP STORE and TurboStore, but differs from BackPack, which always restores spool files with an output priority of 0.) The **spoolpri** keyword lets you specify an output priority overriding that which the files had at the time they were stored.

spoolpri can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the desired **select**.

Example

In the command below, spool files are restored with an output priority of 4.

```
reload from tape spoolpri 4
```

Used with

`reload` `restore`

to

Specifies the device or virtual tape file to which RoadRunner writes the backup.

Syntax

to *destlist* [**and** *destlist*]...

where *destlist* is one or more of the following items, separated by commas. Devices in a destlist are written to sequentially.

**file*	Back-references an MPE/iX file equation.
devdescr	A logical device number or class, plus optional file name and user label information.

Pro Module

where and indicates that the sources it joins are to be written in parallel (Pro Module only.)

where *devdescr* is one of the following:

dev#
or
devclass
or
(dev#|*devclass* **[name** *name*] **[label [ansi|rr]][volid** *[id]*]
[**expdate** *mm/dd/yy*] **[density** *den*])

where:

name	The name of the tape file or virtual tape file. If this option is used, no file equation is required.
label	The type of label, either **ansi** or **rr**.
volid	The volume ID. Up to 6 characters.
expdate	The date after which this tape can be overwritten. In the date format (dd/mm/yy), any number in the yy place holder from 00-27 implies the years 2000-2027. If the yy place holder is 28-99, then the years would be 1928-1999.
density	The density of the source tape in BPI: 800, 1600, or 6250.

Comments

To reduce back up time, RoadRunner can write to multiple devices, either sequentially or in parallel. The devices in a sequential pool are written to one at a time. When one reel completes, RoadRunner begins writing to the next device defined in the destlist. This eliminates the time required to rewind the tape and mount the next reel. When listing devices to be written to sequentially, separate them by commas.

The devices defined in a parallel pool (Pro Module only) are written to concurrently. Writing to tapes in parallel can improve performance when the processing speed of your CPU outstrips the transfer rate of slow storage devices like DAT drives. When listing devices to be written to in parallel, separate them by the word and.

When multiple devices are to be written to sequentially, only the first device in the pool can be qualified with label and density information. Subsequent devices inherit these characteristics.

When parallel tape sets are created, RoadRunner assigns each tape set its own set number. If you do not use ANSI labelling, sets are numbered in the order in which the drives are specified on the to keyword.

About labels:

- If the **label** option is not specified, but a **volid** is specified, or if **label** is specified without a label type, ANSI labeling is assumed.
- If a tape library management system is in use, it automatically supplies a volume ID.
- If you specify a label with either a file equation or the *devdescr* parameter but fail to include a **volid**, and you have no tape management facility on your system, the tape is automatically labeled `$SCR`.
- Before you use ANSI or RoadRunner labels, read "Labeled Tapes" on page 3-13.

to can only be specified once in a command. Use commas or **and** to write to multiple devices.

Examples

To write to a single device defined in a file equation:

```
to *t
```

To write sequentially to three devices defined by file equations:

```
to *t1, *t2, *t3
```

To write to two devices in device class `TAPE` in parallel:

```
to *t1 and *t2
```

To write to a single device, with tape file name t, ANSI label ID 000069, expiration date 6/30/96, and a density of 1600 BPI:

```
to (16 name t volid 000069 expdate 6/30/96 density 1600)
```

Used with

```
copy     full    incremental   interim
partial  store
```

tree/notree/usename

Specifying **tree** initiates a recursive search (the same as appending a slash to the end of your file specification path name).

Specifying **notree** cancels any recursive search (the same as not appending a slash to the end of your file specification path name).

Specifying **usename** (the default) allows the syntax of the fileset specification to determine whether selection is recursive or not (depending on whether a slash is appended to the end of the path name)

Syntax

```
tree
notree
usename
```

Comments

A recursive search is one that selects all files that are descendants of the specified directory. A non-recursive search selects **only** the files in the specified directory.

The **tree**, **notree** and **usename** keywords can be specified globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

Examples

In the following example, **tree** is used to select all files in the Dallas directory, and all of its subdirectories.

```
select /Dallas tree
```

The same selection process can be accomplished by specifying the following:

```
/Dallas/
```

In the following example, all files in the Travis directory are selected, and all subdirectories except the Austin directory and its descendants.

```
select /Travis/(excluding /Travis/Austin/)
```

Used with

full	incremental	interim	partial
reload	restore	store	

trim/notrim

The **trim** keyword releases allocated but unused space that lies beyond the end of a file and returns it to the system as free space. The **notrim** keyword disables this command and is the default.

Syntax

```
trim
notrim
```

Comments

Due to the way file space is allocated, many MPE iX files are built using more space than is actually required for the data in the file. To return that space to the system without changing the file's potential maximum size, use the **trim** keyword when restoring files. To do this, the **trim** keyword invokes an option of MPE iX's :FCLOSE intrinsic.

IMAGE database files reserve all space to the file limit, and cannot be trimmed.

The **trim** keyword can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the desired **select**.

Example

In the following example, the files in myacct are restored with the **trim** option:

```
select @.@.myacct trim restore from tape
```

Used with

reload restore

unlock

Specifies when RoadRunner resets the store bits after files are backed up.

Syntax

```
unlock [byfile | byvolume | atend]
```

where:

byfile	Allows system users access to each file as soon as it has been backed up.
byvolume	Unlocks files when the media volume that the files were written to is completed. This is the default.
atend	Keeps all files locked until the entire backup is complete.

Comments

When files are selected for a backup, RoadRunner sets their store bits to block write access until the backup of each file is complete. The **unlock** keyword allows you to specify when files are released for user access. The **byfile** option is the most convenient for system users, but makes recovery impossible in case of a backup media failure. **byvolume** is the default and is the best compromise in speed, user accessibility, and recoverability. **atend** provides the greatest gain in speed and recoverability, but is least convenient for users.

During online backups, store bits are not set to prevent access. For online backup, **unlock** controls when the files are released from logging. The default, **byvolume**, logs files until the media volume they are written to is complete.

If **unlock** is not specified, RoadRunner resets the store bits **byvolume** for tape backups, and **atend** for virtual tape backups.

unlock can only be specified once in a command.

Example

The following command resets the store bits of each file as it completes.

```
select @.pub.sys store to 17 unlock byfile
```

Used with

```
full   incremental   interim   partial
store
```

volclass

Attempts to place files restored in the specified volume class, overriding and replacing volume class restrictions in the file label.

Syntax **volclass** *volclassname*

Comments When the **volclass** keyword is specified without the **volset** keyword, the volume class is assumed to be contained in the system volume set. If not, the command quits with an error message.

To restore files to a volume class within a volume set other than the system volume set, specify both the **volclass** and **volset** keywords. **volset** *volsetname* selects a specific volume set; volset @ means any volume set.

A file must always be restored into its correct group, regardless of your volume and volume set specifications. If a volume class with the *volclassname* you specified exists in the volume set to which the file's group is bound (even if it is not the system volume set or other volume set specified), the file is restored into that volume class. If no volume class with the name specified exists in the volume set to which the file's group is bound, the file is allowed to span all volumes of its group's home volume set. In this event, any volume restriction in the file label is replaced by a volume set restriction.

Any time a file is restored into a location other than that specified by your use of the **volclass** and **volset** keywords, a warning is printed.

If there is no room in the volume class you specify, the files are not restored and the message "Out of disc space" is issued.

The **volclass** keyword can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

The **volume** and **volclass** keywords are mutually exclusive and cannot be used in the same context.

Examples In this example, RoadRunner attempts to place all files restored onto volumes of the MFG_VOL_CLASS, overriding and replacing the current volume restriction in the file label.

```
restore from 17 select @.@.@
volclass MFG_VOL_CLASS
```

The following command places all files restored onto volumes of MFG_VOL_CLASS in USER_VOL_SET, overriding and replacing the current volume restriction in each file label. If MFG_VOL_CLASS does not exist in USER_VOL_SET, the command quits with an error message.

```
restore from 17 select @.@.@
volclass MFG_VOL_CLASS volset USER_VOL_SET
```

The next command is the same, except volset @ is specified instead of a specific volume set name. In this case, if MFG_VOL_CLASS does not exist in the files' home volume set, the files are restored into their groups, spanning all volumes of the groups' home volume set.

```
restore from17 select@.@.@
volclass MFG_VOL_CLASS volset @
```

Used with

reload restore

volset

Attempts to place files in the specified volume set, overriding and replacing volume class restrictions in the file label. If used with **create**, creates entities on the specified volume set.

Syntax

```
volset @
volset volsetname
```

Comments

When you add the **volset** *volsetname* keyword to a restore command *without* either the **volume** or **volclass** keywords, all files restored are allowed to span the specified volume set, overriding and replacing the current volume restriction in the file label.

Files are always restored to their groups. If the **volset** keyword is used, but indicates a volume set that does not contain the file's group, the file is spanned to the volume set to which its group is bound, and a warning message is displayed. Any volume restriction in the file label is replaced by the volume set to which the group is spanned.

When you use **volset** *volsetname* with either **volume** or **volclass**, the specified volume or volume class is sought within *volsetname*.

If there is no disc space available in the volume set you specify with **volset**, the files are not restored and the message "Out of disc space" is returned.

When you use **volset @** in addition to either **volume** or **volclass**, you tell RoadRunner to restore files into the specified volume or volume class in whatever volume set it occurs, and not to quit if the volume or volume class does not exist.

When you use **volset @** without either **volume** or **volclass**, existing volume restrictions in the file label are changed to the name of the volume set to which its group is bound. The restored file is allowed to span all volumes of that volume set.

When you use **volset** with the **create** keyword (without the **usedir** option), RoadRunner creates groups and accounts on the specified volume set. If a group is created, its home volume set is the specified volume set.

The **volset** keyword can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the desired **select**.

Examples

In the following command, **volset** is used without **volume** or **volclass** keywords. All files restored are allowed to span USER_VOL_SET.

```
restore from 17 select @.@.@
volset USER_VOL_SET
```

In the next example, volset is used with **volclass**. If MFG_VOL_CLASS does not exist within USER_VOLUME_SET, the command quits with an error message:

```
restore from 17 select @.@.@
volclass MFG_VOL_CLASS volset USER_VOL_SET
```

In the next example, **volset @** is used with **volclass**. Here, if volume classes with the name COST_VOL exist in more than one volume set, the home volume set of each file's group determines which COST_VOL it goes into. If any file belongs to a group whose home volume set has no such volume class (or no such volume, if **volume** is specified), it is restored to its correct volume set and a warning is printed. Volume restrictions in the file labels are replaced with a volume set restriction. If COST_VOL does not exist, files are restored into their home volume sets, spanning all volumes. This contrasts with the operation of the same command without **volset** or with a specific **volsetname**; in those cases, if COST_VOL did not exist, the command would fail.

```
restore from 17 select @.@.@
volclass COST_VOL volset @
```

In the following example, groups in myacct are created on my_vol; my_vol becomes their home volume set.

```
restore from 17 select @.@.myacct create volset my_vol
```

Used with

`reload` `restore`

volume

Attempts to place restored files on a specified volume, overriding and replacing volume class restrictions in the file label.

Syntax **volume** *volumename*

Comments When the **volume** keyword is specified without the **volset** keyword, the volume is assumed to be a member of the system volume set. If it is not a member of the system volume set, the command quits with an error message.

To restore files to a volume within a volume set other than the system volume set, specify both **volume** and **volset**. A file must always be restored into its correct group, regardless of your volume and volume set specifications. If you use both **volume** and **volset**, and the specified volume exists in the file's home volume set (even if it is not the system volume set or other volume set specified), the file is restored on that volume. If no volume with the name specified exists in the volume set to which the file's group is bound, the file is allowed to span all volumes of its home volume set. In this event, any volume restriction in the file label is replaced by a volume set restriction.

Any time a file is restored into a location other than that specified by your use of the **volume** and **volset** keywords, a warning is printed.

If there is no available disc space in the volume you specify, the files are not restored and the message "Out of disc space" is returned.

The **volume** keyword can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

The **volume** and **volclass** keywords are mutually exclusive, and cannot be used in the same context.

Example The following command attempts to place all restored files on VOLUME_15. If VOLUME_15 does not exist in the system volume set, it quits with an error message:

```
restore from 17 select @.@.@ volume VOLUME_15
```

The next command places all files in the restore fileset on VOLUME_15 in the ACCTG_VOLUME_SET. If VOLUME_15 does not exist in the ACCTG_VOLUME_SET, the command quits with an error message:

```
restore from 17 select @.@.@
volume VOLUME_15 volset ACCTG_VOLUME_SET
```

The next command is the same except **volset @** is specified instead of a specific volume set. In this case, if VOLUME_15 does not exist, the files are restored into their groups, spanning all volumes of the groups' home volume sets.

```
restore from 17 select @.@.@
volume VOLUME_15 volset @
```

Used with

reload restore

where

Selects files by their attributes.

Syntax **where** *expression*

where *expression* can be any of the criteria listed below connected by the logical operators **and**, **or**, and **not**. Additionally, parentheses may be used to indicate the order of evaluation. See Examples below.

accdate *relop date* [**at** *hh:mm*]	Access date and time.
acctime *relop hh:mm*	Access time.
blockfactor *relop value*	Blocking factor.
blockfactor in *range*	Range of blocking factor.
blocksize *relop value*	Block size.
blocksize in *range*	Range of block size.
creator *creatorname*	Creator name.
credate *relop date* [**at** *hh:mm*]	Creation date and time.
cretime *relop hh:mm*	Creation time.
eof *relop value*	End of file (indicates file size).
eof in *range*	End of file, range of file sizes.
filecode *relop value*	File code.
filecode *in range*	Range of file codes.
hasacd	Whether ACDs are used.
hasctl	Whether file has carriage control.
isascii	Whether data in file is ASCII.
isbinary	Whether file is binary.
iscir	Whether file is circular.
isfixed	Whether file has fixed length records.
isimage	Whether file is part of an IMAGE database.
isksam	Whether file is a compatibility mode KSAM file.
isksamxl	Whether file is a native mode KSAM file.
ismsg	Whether file is a message file.
isrio	Whether file is an RIO file.
isspool	Whether file is a spool file.
issql	Whether file is part of an SQL database.
isstd	Whether file is a standard MPE/iX flat file.
isundefined	Whether file is of undefined record lengths.
isvariable	Whether file is of variable record lengths.
limit *relop value*	The file's limit.
limit in *range*	Range of file's limit.
moddate *relop date* [**at** *hh:mm*]	Modification date and time.
modtime *relop hh:mm*	Modification time.

numextents *relop value*	Number of extents in file.
numextents in *range*	Range of extents in file.
numlabels *relop value*	Number of user labels in file.
numlabels in *range*	Range of user labels in file.
privlev *relop value*	Privilege level of file.
privlev in *range*	Range of privilege levels of file.
recsize *relop value*	Record size of file.
recsize in *range*	Range of file record sizes.
space *relop value*	Number of sectors in file.
space in *range*	Range of file sectors.

where *relop* is any of the following:

```
=, <, >, <=, >=, !=, !<, !>
equal, less than, greater than, less than or equal to, greater
than or equal to, not equal to, not equal to or less than, and
not equal to or greater than.
```

where *range* is :

lowval **[thru** *hival***]** **[,***lowval* **[thru** *hival***]]...**

where *lowval* and *hival* are beginning and ending numeric values that define a range.

where *date* is:

datespec **[±** *numberofdays***]**

where *datespec* is:

lastfull	The date of the last full backup.
lastpart	The date of the last partial backup.
lastinterim	The date of the last interim backup.
today	The current system date.
mm/dd/yy	Month, date and year.

Comments

You can specify a virtually unlimited number of criteria, using the logical operators above in any combination. Add parentheses to control the order in which the expression is evaluated; expressions in parentheses are evaluated first. Multiple **where** statements, either local or global, have a cumulative effect, just as if they were connected by **and**. For example, if a local **where** statement selects files by the **not isascii** expression and a global statement uses the **isascii** expression, no files will be selected by the local **select** statement. This illustrated in the Examples section.

The **where** keyword can be specified both globally (in relation to the entire command) and locally (in relation to a selected set of files). To specify a keyword locally, place it within the parentheses following the **select**.

where can also be used in **excluding** expressions.

Examples

The following example uses `where` to limit the selection to files with over 1000 records, whose record sizes are either 72, 80, 128.

```
where eof > 1000 and recsize in 72, 80, 128
```

This command limits the selection to files with over 1000 records and record sizes between 72 and 128:

```
where eof > 1000 and recsize in 72 thru 128
```

Here, selected files are limited to those that are not ASCII files with record sizes of either 72 or 80 bytes:

```
where not (isascii and (recsize = 72 or recsize = 80))
```

In the following example, **where** is applied both locally and globally. The expressions enclosed in parentheses with a **select** apply only to the files listed within the parentheses, while the **where** statement outside the parentheses applies to the command as a whole.

```
select (@.pub.dev where filecode in 1 thru 100)
select (@.oub.gl where filecode != nmprog and (isascii or iscir))
where space < 100000 and moddate in 1/1/95 thru 1/1/96
```

To select files created after five p.m. on May 5, 1996:

```
where credate > 5/5/96 at 17:00
```

To select files created after five p.m. on any date, as well as files created since May 5, 1996:

```
where credate > 5/5/96 or cretime >17:00
```

If contradictory local and global criteria are specified, they cancel each other out. This is illustrated by the example below:

```
select (@.pub.sys where isascii)
select @.@.telsup
where not isascii
```

This command would select no files from pub.sys since the combined local and global criteria for this group indicate that files are to be selected only if they are both ascii and not ascii. In other words, it is the same as entering:

```
select (@.pub.sys where isascii and not isascii)
select (@.@.telsup where not isascii)
```

The correct way to express the intended selection criteria is:

```
select (@.pub.sys where isascii)
select (@.@.telsup where not isascii)
```

Used with

copy	full	incremental	interim
listdir	partial	reload	restore
scan	store	validate	

Appendix

A

BackPack/XL and TurboStore Compatibility

This appendix provides special information for upgrading BackPack/XL, HP STORE, and TurboStore II installations. BackPack/XL is only supported for RESTORES on MPEiX 5.0 and later.

Note

RoadRunner and BackPack/XL cannot read each other's tapes, as the two products use different tape formats. Files must be restored using the software with which the backup was created.

Interpreting Commands

To have RoadRunner interpret BackPack/XL, HP STORE, and TurboStore II commands, run the program with `PARM=1`. For example:

```
:run rr.pub.tym;parm=1
RoadRunner 4.5.7 * Copyright 1992-1996 Unison Software *Tue
13Jan96 7:14pm
>
```

The RoadRunner program banner is displayed, but the BackPack/XL greater-than prompt is displayed instead of the standard prompt to show that you are in compatibility mode. You can enter commands exactly as if you were running BackPack/XL , HP STORE, or TurboStore II. For keywords with no direct equivalents, RoadRunner substitutes the options shown in the table on page A-3.

You cannot switch between compatibility mode and RoadRunner's normal command interpreter. To enter RoadRunner commands, exit and run the program again without `PARM=1`.

Translating Commands

To translate a BackPack/XL, HP STORE, or TurboStore II command to a file containing RoadRunner syntax, run RoadRunner with `PARM=1` as shown in the previous section. Type the complete text of the command, then add the **translate [=filename]** keyword. The command is translated (see "Syntax Tables," on page A-3 for equivalent syntax) and placed in the file indicated by the **filename** parameter. If no filename is specified, the command is written to a file called `RRXLATE` in your logon group and account. An example of this is shown below:

```
:run rr.pub.tym;parm=1
>store @.@.@;*t;setbpdate;compress=6;dbstore;defer;&
```

```
>progress=5;directory;translate=mystore
>exit
```

The file mystore would contain the following:

```
SELECT @.@.@
FULL TO *T
COMPRESS 2 DBSTORE *
DISPLAY PROGRESS 5
DIRECTORY
```

The translated file is not executed automatically. To execute it, run RoadRunner, text the file, and type go.

Note

POSIX filename specifications in translate mode are not supported.

If the store fileset specified on the original command is an indirectfile (!filename), RoadRunner checks the file to see if it contains only fileset specifications, or fileset specifications and other keywords. If it includes only fileset specifications, it can be used as is. If it contains other keywords, RoadRunner translates the indirectfile into an include file. First, it prompts you for the name of an include file to create. If you press return instead of specifying a file name, it prompts you for the name of an existing include file to use. If you press return again, RoadRunner translates the information in the indirect file and places it directly into the translated command. To read more about indirectfiles and include files, see page A-5.

Translating Job Streams

The most convenient way to translate existing job streams is to follow the steps below:

1. Use any text editor to modify the job stream so that it contains only the actual backup commands and add the **translate** keyword at the end. Save it with another name, for example, myfile.
2. To translate this file, type the following:

   ```
   :run rr.pub.tym;parm=1
   >use myfile
   >exit
   ```

3. Run your text editor again. Text in your original jobstream and do the following:
   - Replace the `:RUN` command that executes your old backup program with `:run rr.pub.tym`.
   - Delete the lines containing the old backup commands.
   - Add the lines from **rrxlate** in their place.
   - Place `/go` on a new line following the last translated line.
   - Add the RoadRunner **exit** command at the end of the translated lines.
   - Keep the file.
4. The job is now ready to stream.

Syntax Tables

The following table compares keywords used with BackPack/XL, HP STORE and TurboStore to those used with RoadRunner. Keywords that are exactly the same in RoadRunner as they are in the other products (for example, VOLSET=) are not included. In addition to indicating how keywords are handled in the interpretation and translation processes described above, this table should be of use to users who are learning RoadRunner syntax.

BackPack or TurboStore keyword	RoadRunner equivalent
`ACCOUNT=xxx`	`change account @ to xxx`
`BPDEFER file equation`	`after all "stream *bpdefer"`
`COMPRESS=value`	`compress 1` for values 1, 2, 3, and low `compress 2` for values 4, 5, 6, and high
`CREATOR=xxx`	`change creator @ to xxx`
`DATE>=date`	`where moddate >= date`
`DATE<=date`	`where accdate <= date`
`DATE=date`	`where credate = date`
`DEV=xxx`	`vol xxx` if the parameter is numeric or `volclass=xxx` if the parameter is alphanumeric.
`DIRECTORY`	For `restore`: `directory update, udcs`
`FASTDIR`	Ignored.
`FCRANGE=xxx/yyy`	`where filecode in xxx thru yyy`
`fileset`	`select fileset`
`fileset1-fileset2`	`select fileset1 excluding fileset2`
`fileset(DATE>=date)`	`select (fileset where moddate >= date)`
`FILES=xxx`	Ignored.
`GETBPDATE`	**full date**
`GROUP=xxx`	`change group @ to xxx`
`!indirectfile`	`^indirectfile`
`INTER`	Ignored.
`LABEL=`	`to|from (dev label rr ...)` Any text in the **string** portion of the BackPack/XL label becomes a quoted string of the **description** keyword.
`LINKSPF`	Ignored; spoolfiles are always linked by RoadRunner
`MAXTAPEBUFF`	Ignored.
`MEMBUFF`	Ignored.
`MINMEM`	Ignored.
`MOSET`	Ignored, produces error.

BackPack or TurboStore keyword	RoadRunner equivalent
NAME	Used only with optical disk by HP utilities. Ignored, produces error.
ONERROR	ONERROR=REDO is RoadRunner's default and is Ignored. as there is no equivalent keyword. ONERROR=REDO is translated to after error quit.
PROGRESS	**display progress**
REPLACE	Ignored.
RESERVE	Ignored.
RESTORE...LISTDIR	Changed to **listdir** command.
RESTORESET	**from**. Parallel and serial device designations are preserved.
SETBPDATE	**full date**
SHOW	**report**
SKIPDIR	**recover media**
SPLITVS	Ignored, produces error.
STORESET	**to** . Parallel and serial device designations are preserved.
TAPES=n	**to *bpauto1,*bpauto2,..,*bpauton**
*storefile	**to *storefile**
TRANSPORT	Ignored, produces error. Use BackPack/XL to transport files to MPE V systems.
VSTORE	**validate**
VSTORESET	**from**. Parallel and serial device designations are preserved.

The following table lists commands used with BackPack/XL's **BPUTIL** utility and their RoadRunner equivalents.

BP-TIL	RoadRunner
COPY	**copy**
DATE	**fulldate, partdate, interimdate**
LABEL	**listinfo**
LISTDIR	**listdir**
PURGE	**vtpurge**
RENAME	**vtrename**

Differences Between RoadRunner and BackPack/XL

Handling of Virtual Tape Files

There are several differences between the ways BackPack/XL and RoadRunner handle virtual tape files. The filecode for RoadRunner virtual tape files is -21074, whereas with BackPack/XL it was -16976. Each file is built using default file characteristics (256 byte records, binary, 1 record per block). Extents are allocated as needed, so there is no need for the RESERVE keyword.

Each new virtual tape backup purges any existing backup with the same name (REPLACE always in effect). The file name of the first virtual tape file RoadRunner creates is always the name of the "root" with no numeric digits appended and up to 8 characters in the name. The second file becomes root02, etc. BackPack/XL requires 6 or less characters and always appends "01" to form the name of the first virtual tape file. The complete directory is written only to the first virtual tape file. In addition, the restore only version of RoadRunner (**roadrest**) is not placed in the first virtual tape file as it is by BackPack/XL. This greatly reduces the space and time required for small virtual tape stores.

While BackPack/XL did not store its own virtual tape files, RoadRunner does store virtual tape files (whether created by BackPack/XL or RoadRunner) unless they are expressly excluded. To exclude them, use a `select` statement similar to the ones shown below:

```
select @.@ where filecode<>-21074 (excludes RoadRunner VT files)
select @.@ where filecode<>-16976 (excludes BackPack/XL VT files)
select @.@ where filecode<>-21074 and filecode<>-16976
(excludes RoadRunner and BackPack/XL VT files)
```

Output Priority of Restored Spool Files

By default, RoadRunner restores spool files with the same output priority they had at the time they were stored. This matches the defaults of HP STORE and TurboStore, but differs from BackPack/XL, which always restores spool files with an output priority of 0. To override the stored output priority see the **spoolpri** keyword on page 5-68.

IndirectFiles

In order to provide more flexibility when using indirectfiles, the format of RoadRunner indirectfiles differs from BackPackXL indirectfiles. With RoadRunner, only filenames are allowed in the indirect files. This eliminates the ambiguity of having excludes in an indirectfile which itself was specified with the **excluding** keyword in a **select** statement.

RoadRunner indirectfiles can have multiple filenames per line, as long as they are separated be at least one space. Comments, delimited by curly braces ({ and }), are allowed anywhere in the file and can span multiple lines.

If your existing BackPack/XL indirectfiles contain only filenames, there is no need to convert them. A better solution for files that contain excludes or date specifications is to convert them to RoadRunner include files.

An include file can contain any RoadRunner keyword and generally provides better performance than an indirectfile. The only reason to use indirectfiles is if there are so many files in the list that they don't fit in RoadRunner's internal command buffer.

You can convert a BackPack/XL indirectfile to a RoadRunner include file as follows. This BackPack/XL file:

```
@.@.accting
@.@.mfg(date >=11/15/96)
-@.test.@
-@.archive.@
```

can be converted to this RoadRunner file:

```
select @.@.accting
select (@.@.mfg where moddate >=11/15/95)
excluding @.test.@,@.archive.@
```

You then save this command in a file called selects.backup and use it as follows:

```
:run rr.pub.tym
<01>include selects.backup
<02>partial to tape
<03>report
<04>/go
```

Restoring Files on an MPE/iX System

Files from a BackPack/V STORE tape can be restored on a Precision Architecture HP 3000. A copy of the BackPack/XL restore module, BPXLREST, is provided with BackPack/V for this purpose. Restore BPXLREST onto any Precision Architecture HP 3000 with the MPE/iX :RESTORE command; then run BPXLREST to restore files from the BackPack/V or BackPack/XL tape. BPXL is also provided on the RoadRunner product tape. After installing RoadRunner (See the installation instructions in the *Getting Started Guide*), run BPXL.PUB.TYM to restore files from a BackPack/V or BackPack/XL tape.

Appendix

B Utilities

RoadRunner is shipped with several utility programs that are installed on your system when you restore the product tape. The utilities include:

- **AUTOREP**: Configures tape drives for auto-reply.
- **BPXL**: The current restore module of BackPack/XL.
- **XLOGOFF**: Logs users off prior to backup.

Each of these is discussed in this Appendix.

AUTOREP

AUTOREP.PUB.TYM converts a tape drive or serial device from normal manual reply mode to auto reply mode before you run an unattended backup. It is used again to convert the device back to normal reply mode after backup is complete. (For nonremote backups, it is usually simpler to use the **autoreply** keyword on page 5-8.)

To convert your device to auto reply mode, run **AUTOREP** as shown, where n is the logical device number of your tape drive:

```
:RUN AUTOREP.PUB.TYM;PARM=n
```

To convert the device back to normal reply mode, run **AUTOREP** again, setting the **PARM** parameter equal to the **LDEV** number of your tape drive, preceded by a minus sign:

```
:RUN AUTOREP.PUB.TYM;PARM=-n
```

If you have multiple tape drives, run **AUTOREP** as many times as necessary. The best way to use this utility is to embed these commands in the job stream that performs your backup. Be sure to specify your tape drive's specific **LDEV** number in the file equation for your tape file. This avoids confusion if more than one device is configured as type **TAPE**. The **LDEV** number you use in the **PARM** keyword is the same as the one referenced in the file equation for your tape file. In this sample job stream, the **LDEV** number of the tape drive is 7:

```
!JOB JBACKUP,OPERATOR.SYS
!FILE T;DEV=7
!RUN AUTOREP.PUB.TYM;PARM=7    (Convert to auto reply)
!RUN RR.PUB.TYM
SELECT @.@.@ STORE TO *T REPORT
/GO
EXIT
!RUN AUTOREP.PUB.TYM;PARM=-7  (Convert to normal reply)
!EOJ
```

For the **AUTOREP** program to execute successfully, your tape drive must not be in use by another process. If the device is not available, **AUTOREP** aborts with this message, followed by one or more lines of diagnostic information:

```
Device is not at BOT or not a SERIAL device
ABORT: AUTOREP.PYB.TYM.%0.%105
```

If the logical device number you specify is not a tape drive or serial device, **AUTOREP** aborts with this message:

```
Invalid device specified
ABORT: AUTOREP.PYB.TYM.%0.%157
```

BPUTIL

The **BPUTIL** program is provided to allow former users of BackPack/XL to perform the following functions on their existing BackPack/XL-created backups:

- Copies virtual tape files to magnetic tape.
- Copies backups from tape to tape.
- Purges virtual tape files.
- Renames a complete virtual tape fileset with one command.
- Prints the BackPack/XL label (indicates creation date and reel number) for a tape reel or virtual tape file.
- Prints file directory listings.
- Alters the date and time of backup stored by the **SETBPDATE** keyword of the BackPack/XL **STORE** command.

A **REDO** function is also provided. See the *BackPack/XL User Guide* for details about **BPUTIL**.

BPXL

This is a copy of the most current restore module of BackPack/XL, compatible with the current release of MPE iX. It is provided for the convenience of former BackPack/XL users. See the *BackPack/XL User Guide* for details on restoring files.

XLOGOFF

The **XLOGOFF** utility logs off all users in preparation for a backup by issuing a series of **ABORTJOB** commands. For any session being logged off, an **ABORTIO** command is issued for the session's logical device before the **ABORTJOB**. This is required for users running **VPLUS** or other block mode applications.

Setting Up Exclusions

To specify logical devices and/or logon IDs that are not to be logged off, place entries in a file called `TYMEXCLD.PUB.SYS` with any editor. Use the following kinds of entries.

`#nn`	A logical device, such as `#20` or `#101`
`#Jnnn`	A job number, such as `#J@` or `#J24`. `@` can be used for all jobs.
`#Snnn`	A session number, such as `#S@` or `#S334`. `@` can be used for all sessions.
`$state`	An executing state, such as `$WAIT` or `$SUSP`

`job,user.acct`

Any logon ID, such as JOE,@.@ or @.SYS. If no job name is specified, all jobs associated with the user are skipped.@,OPER.SYS is the same as OPER.SYS. @ can be used for all jobs/sessions, all accounts or all users.

All entries are terminated by the first space, so there should be no embedded spaces. After the first space, comments can be added. If you use EDITOR's default record length of 80 bytes, about 75 records can be accommodated in the file. If you need more records, create a file with a shorter record length.

Following is a sample file including comments:

```
#20             «to skip the console»
$SUSP           «to skip suspended jobs»
@.SALES         «to skip sales staff, job or session»
OPERATOR.SYS    «to skip the system operator and his jobs»
@.HPMAIL        «to skip the electronic mail jobs»
DNG,@.@         «to skip jobs/sessions named 'DNG'»
```

Running XLOGOFF

Before running **XLOGOFF**, you must do one of the following:

- Enter the following command from the system console:

  :ALLOW @.@;COMMANDS=ABORTJOB,ABORTIO

  This gives everyone on the system the ability to abort any job or session and remains in effect until the next system start-up or until the following command is entered:

  :DISALLOW @.@;COMMANDS=ABORTJOB,ABORTIO

- Run Boeing's **ALLOWME** program from the INTEREX CSL tape.

Once you allow the procedure and set up the exclusions file, `TYMEXCLD.PUB.SYS` (this file must exist for **XLOGOFF** to run, even if there are no records in it), **XLOGOFF** can be executed by any user allowed this capability:

```
:RUN XLOGOFF.PUB.TYM
```

To prevent **XLOGOFF** from terminating the job or session running it, the user's own job/session number is automatically added to the exclusion list. Scheduled jobs are always skipped by **XLOGOFF**. This is done to avoid unintentional interference with critical system routines. If any scheduled job needs to be logged off prior to backup, the operator must take care of it manually.

Trace Facility

A trace facility is provided to allow you to test the operation of the program and any entries placed in the exclusion file. When in trace mode, the **ABORTIO** and **ABORTJOB** commands are not actually sent, but are printed with an indicator showing how the program deals with the job/session. Trace mode is triggered by setting the JCW **XLOGOFFTRACE** to any non-zero value before running **XLOGOFF**, as shown below:

```
:SETJCW XLOGOFFTRACE = 1
:RUN XLOGOFF.PUB.TYM
```

Version Number

If you call our technical support representatives with questions about **XLOGOFF**, they need to know which version you are running. To display the version number, enter the following:

```
:RUN XLOGOFF.PUB.TYM,VERSION
```

Appendix

C

Technical Information

CI Variables

The following command interpreter variables are set by RoadRunner during store, restore, and other operations and maintained by MPE iX. They are listed in alphabetical order.

CI Variable	n=
`rrcancelled`	The number of files selected, which RoadRunner was unable to process because of an error in the store or restore operation. These files were originally included in **rrselected**.
`rrcompression`	Compression factor achieved during store.

`rrcpuprocess1`
`rrcpuprocess2`
`...`
`rrcpuprocessn`
	For tape backups, the number of CPU processes is equal to 3 times the number of parallel paths used plus 1. The value of each variable is equal to the number of CPU milliseconds used by the particular process.

`rrcpuinputrate`	Average kilobytes per CPU second read from disc.
`rrcputime`	CPU milliseconds required by all processes combined.
`rrelapsedtime`	Duration of RoadRunner operation in milliseconds.
`rrerror`	The number of the last error to occur.
`rrmediainputrate`	Average kilobytes per second read from disc. Only seconds during which tape writes occur count toward this total. Time spent waiting for operator intervention is not included.
`rrmediaoutputrate`	Average kilobytes per second written to the output medium. Only seconds during which tape writes occur count toward this total.
`rrnotfound`	Number of files selected for restore or validate but not found on medium. This setting may be unusually high when you use the **single** keyword or **validate** function.
`rrrejected`	Number of files rejected during directory build, for example if the storebits were already set by another process, or if the file is not archivable because the "nobackup" bit was set in the file label. This count is not included in **rrselected**.
`rrrestored`	Number of files restored.
`rrsectorsexpected`	Number of uncompressed sectors to be processed.
`rrsectorslogged`	Number of sectors used to log transactions or apply logged transactions during online backup or restore.

CI Variable	n=
`rrsectorsneeded`	If you run out of disc space while restoring files, this variable indicates the number of sectors needed to restore the unrestored portion of the fileset.
`rrsectorsread`	Number of sectors read from disc before compression, excluding file labels.
`rrsectorswritten`	Number of sectors written to the output device.
`rrselected`	Number of files selected by the directory build.
`rrstored`	Number of files stored.
`rrversion`	The numeric version number. The lower 3 bytes indicate the major version, minor version and test version numbers respectively.

Abbreviations/Reserved Words

RoadRunner's reserved words include utility functions, commands, keywords, and keyword options. To make entering commands easier, all RoadRunner reserved words can be abbreviated, as long as they remain unique from all other RoadRunner reserved words. The following alphabetical listing can help you determine how many letters are required to make a reserved word unique. For example, **dbr** could be entered for **dbrestore**, as that differentiates it from **dbfast** and **dbstore**.

1600	2up	4up	6250
800	accdate	account	accountname
acct	acctime	add	after
all	and	ansi	append
at	attend	autoeject	autoload
autoreply	backupname	blockfactor	blocksize
bs	buffer	byfile	byvolume
change	check	code	combine
command	compress	compression	concurrent
copy	copyacd	create	creator
credate	cretime	cs	datefile
dates	dbfast	dbrestore	dbstore
delete	density	description	directory
display	do	ds	duplex
each	environment	eof	error
errors	es	exclude	excluding
exit	expdate	extents	file
filecode	filename	files	format
from	full	fulldate	fullname
go	group	groupname	hasacd
hascctl	header	help	ignlabel
ignorelabel	in	include	incremental
index	interim	interimdate	interleave
isascii	isbinary	iscir	isfixed
isimage	isksam	isksamxl	ismsg
isrio	isspool	issql	isstd
isundefined	isvariable	keep	keepnew
label	landscape	lastfulldate	lastinterimdate
lastpartdate	limit	lines	list
listdirectory	listinfo	local	lock
lockword	long	lp	maxerrs
media	mediaswitch	mediavolume	minimum
moddate	modify	modtime	mpe
name	new	newdate	nocompress
nocopyacd	nodbfast	nodbrestore	nodbstore
nodirectory	nofiles	noignlabel	noignorelabel
nokeep	nolist	nolocal	nolock
noonline	nopurge	norejects	normal
noroadrest	not	notonvs	notrim
nousedir	null	numextents	numlabels
offline	olddate	online	onvs
or	ospfname	partdate	partial
portrait	priority	progress	purge
purgeafter	purgebefore	recover	recsize
rectype	redo	reload	report
restore	rr	scan	sectors
security	select	short	single
space	splitvs	spoolpri	statistics
stats	stdlist	store	syslist
text	thru	to	today
trim	udcs	unlock	update
usedir	validate	volclass	volid
volinfo	volname	volnames	volset
volume	vtpurge	vtrename	warnings
where	width	with	

RoadRunner Support for POSIX

RoadRunner now features POSIX Hierarchical File System (HFS) capability, in addition to its support for standard MPE syntax. Newer versions of MPE/iX allow, but do not require, use of the POSIX HFS. The following sections provide a brief overview of POSIX directory structures and changes in RoadRunner to support Hierarchical File Systems.

If you do not want to incorporate the HFS directory structure with newer versions of MPE/iX:

- Do not modify your directory structure to include HFS directories
- Do not run applications that adhere to the POSIX HFS standard

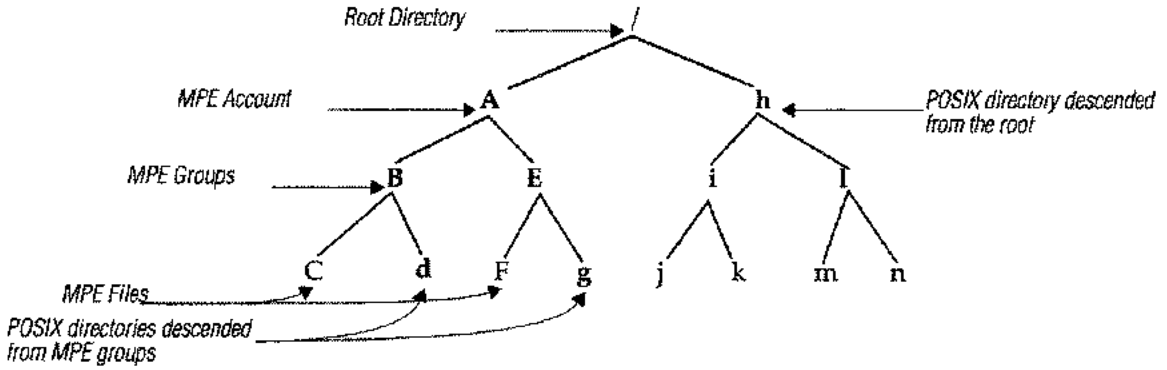
If you do have POSIX directories on your system, or want to take advantage of the increased flexibility that HFS directory structures offer, read the following sections for information about RoadRunner's POSIX support.

Complying with these restrictions ensures that you will not need to worry about any of the POSIX-related changes within RoadRunner or the operating system. Any RoadRunner commands you have written in the past still back up the same files they always have.

For detailed information about the POSIX HFS, refer to Hewlett-Packard's *New Features of MPE/iX: Using the Hierarchical File System* (part no. 32650-90351). Introduction to POSIX

The following diagram illustrates a sample MPE/iX directory structure. Upper case letters represent MPE structures, while lower case letters represent POSIX structures

In the HFS structure, MPE directories MUST descend from the root (/). You cannot have MPE directories or files that descend from a POSIX directory. However POSIX directories and files can descend from an MPE directory.



Note

Under POSIX, all directories, including MPE (non-hierarchical) directories are **subsets** of the HFS root, indicated by a slash (/). Use a slash (/) to select all files. If you use @.@.@ to select all files, a message will be displayed indicating that this entry has been interpreted as a slash (/). For more information on wildcards and their use, see the *Quick Start Guide*.

The HFS allows you to maintain MPE directory structures within the POSIX superset. This capability allows you to operate within both POSIX and MPE *namespace* (refer to the

"Glossary of POSIX Terms" below) which are not exclusive, but have different characteristics.

The following conventions apply to HFS:

- Within MPE namespace (which sees only entities with MPE syntax), MPE upshifts all characters entered, and files are specified in traditional MPE syntax:

  `FILE.GROUP.ACCT.`

- Within POSIX namespace, (which sees both MPE and POSIX entities), POSIX is a superset of MPE namespace. The POSIX namespace provides the following unique features:
  - Case sensitive file names. File, directory and group names are no longer automatically upshifted. This means that `FILE` and `File` are two different entities.
  - HFS syntax. Sometimes referred to as *tree* structure syntax. Under HFS, files are typically specified from the root (`/`) down: `/ACCT/GROUP/FILE.`
  - In HFS, files can also be specified in reference to the current working directory by using a dot (`.`) to refer to the current directory, or `..` to refer to the current directory's parent directory.
  - Unlimited directory nesting. Directories can be nested arbitrarily deep.
  - 256 mixed case character file or directory names. If the directory or file is directly under the root (`/`), or located within an MPE group, the name is limited to 16 mixed case characters; otherwise, 265 characters is the limit.
  - The limitation on the total path name cannot exceed 1,024 characters.

Glossary of POSIX Terms

The following commonly used terms and characters are used throughout this update, and have special meaning in the POSIX environment. The directory structure referred to in the definitions is shown in the diagram on page 1 of this update.

This glossary is not intended to be all-inclusive: for a more complete listing of POSIX—specific terms, see Hewlett-Packard's *New Features of MPE/iX: Using the Hierarchical File System* (part no. 32650-90351).

pathname The system's directions to the file or directory from the root. A pathname can be absolute (as it is descended from the root) or relative (as it descends from the current working directory).
The absolute pathname for the `m` directory is: `/h/l/m`, while the relative pathname (if you are in the l directory) is `./m`.

component An element in the pathname. In the above directory, `l` is a component in the pathname for the `m` directory.

descendant A file or directory below the current directory in the hierarchical file structure. In the above directory, `m` is descendant from `l`.

parent A directory from which others descend. In the above directory, `l` can be considered the parent of `m`.

usename The default file selection criteria, indicating that the file or directory name as entered is the criteria for selection. See page 5-71 for more information.

recursive search

Specifies the current directory and all its descendants. Recursive searches can be initiated by appending a slash to the end of the directory name in the select statement. In the above directory, a selection criteria of `l/` indi-

cates the `l`, `m`, and `n` directories and their files. See the **tree/notree/username** keyword on page 5-71 for more information.

namespace A way of interacting with the directory structure, to which certain rules apply. If using HFS syntax, you are in POSIX namespace. If you are using the traditional MPE syntax, you are in MPE namespace.
For a description of MPE and POSIX namespace, see the description on page C-4.

spanned This term relates exclusively to volume set management concerns. POSIX directories which descend from an MPE group are created on the same volume set that the group is spanned to. Directories that do not descend from an MPE group are spanned to the root directory (`MPEXL_SYSTEM_VOLUME_SET`). See page C-7 for more information.

/ A slash by itself indicates the root directory, from which all others descend. When appended to the end of a selection criterion, as in `l/`, it indicates a recursive search.

. A dot (.) indicates the current working directory without entering its name.

.. Indicates the parent of the current working directory. In the above directory, specifying `..` while in the `m` directory indicates the `l` directory.

RoadRunner Commands and Keywords Affected by POSIX

RoadRunner's implementation of POSIX filename syntax is directly supported by modifications to the following keywords:

change (see page 5-10)
create (see page 5-20)
directory (see page 5-31)
report (see page 5-55)
excluding (see page 5-33)
select (see page 5-60)

The following keywords support HFS, but their use has not been modified by the new capability:

create full partial incremental interim include

In addition, another keyword has been added to support recursive searches in the new POSIX HFS directory structure. The following keyword combination has been added to support POSIX.

tree/notree/usename (see page 5-71)

The Change Keyword

The **change** keyword is explained in detail on page 5-10.

A new modifier, **path**, has been added to the **change** keyword. The **path** modifier allows a path or set of paths to be changed on either a global or local basis when storing or restoring files. The **creator** option now allows the account name part of the **username** to be changed.

The Create Keyword

The **create** keyword is explained in detail on page 5-20.

A new option for the **optionlist** parameter has been added: **path**. The **path** option creates path, group, and account information for restored files.

The Directory Keyword

The **directory** keyword is explained in detail on page 5-28.

The functionality of the **directory** keyword has been expanded to include POSIX directories. While this added capability has not required any changes in the syntax or use of the **directory** keyword, there are several things to keep in mind when dealing with POSIX directories.

The Report Keyword

The **report** keyword is explained in detail on page 5-55.

To support the printing of POSIX path names, several changes have been implemented in the RoadRunner reporting system. Two new options have been added—the **pathname** option and the **filename** option.

Appendix

D

Error Messages and Warnings

Error and warning messages are listed alphabetically below. Where an exclamation point (!) appears in the error message text, it is replaced with a value. To look up messages by number, type **help error***nn* or **help warning***nn* at the <rr> prompt, where *nn* is the error number.

Error or Warning	Additional Information
! is being stored by another backup process	The store bit in the file specified was already set. This could be due to another store taking place or an overlap in the selected backup filesets.
! is opened for writing	The file is not stored and is listed in the **files not stored** section of the report listing.
! is open exclusively (NOLOCK in effect)	Even with **nolock**, files opened exclusively by another process cannot be stored. The file is not stored and is listed in the **files not stored** section of the report listing.
A backup media error occurred while restoring file !. Since the **recover** option was selected, an attempt will be made to restore as much of the file as possible.	This message, issued in conjunction with the **recover files** options indicates that some of the indicated file is missing due to a backup media error.
A catastrophic I/O error has occurred	A tape error occurred from which RoadRunner is unable to recover. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
A file I/O error occurred while trying to restore file !	An MPE procedure failed while trying to reserve memory or post data for a file being restored. For more information, check the MPE status message printed after this message.
A lockword was specified for virtual tape fileset !, lockwords are not allowed	Lockwords can not be specified as a part of virtual tape file names. Remove lockwords from the **to** or **from** statement, or any file equations that reference virtual tape files and retry the operation.
A media error occurred while reading the directory on device !	The tape drive may need cleaning or repair, or the volume may be corrupt. For more information, check the MPE status message printed after this message. Restart the restore with a different volume of the backup or use **recover media.**
A POWERFAIL was detected on device !	Make sure the device is powered up and that the volume is mounted properly. RoadRunner should continue with the operation when the device is ready.
A section of data is missing a continuation descriptor	There is data corruption in a media block of the backup. Use **recover files** to recover as much of the affected files as possible.
Active data found for an unopened file	There is data corruption in a media block of the backup. Use **recover files** to recover as much of the affected files as possible.

Error or Warning	Additional Information
An error occurred while allocating disc space for virtual tape fileset !	An MPE procedure failed while allocating disc space for a virtual tape file. Check to see that there is enough disc space for your backup and retry the operation. For more information, check the MPE status message printed after this message.
An error occurred while checking ACD access to file !	The access control definition in the label extension for the file is invalid and thus cannot be successfully checked. Access to this file is denied for all users except those with SM or OP capability.
An error occurred while logging online data for file !	An error occurred during an attempt to log information for the indicated file. This may be due to insufficient disc space. For more information, check the MPE status message printed after this message.
An error occurred while printing a report line to !	An error occurred while printing to a report destination. An MPE/XL error message should follow this message which gives more information.
An error occurred while reading file !	For more information, check the MPE status message printed after this message.
An error occurred while reading from virtual tape file !	The MPE FREAD intrinsic failed. For more information, check the file system error message printed after this message.
An error occurred while reading the backup date file !	The MPE FREADDIR intrinsic failed. For more information, check the file system error message printed after this message.
An error occurred while reading the directory from virtual tape file !	The MPE FREAD intrinsic failed. For more information, check the file system error message printed after this message.
An error occurred while reading the label on device !	A tape I/O request failed. For more information, check the MPE status message printed after this message. The tape drive may need cleaning or repair, or the volume may be corrupt. Restart the restore with a different volume of the backup or use **recover media**.
An error occurred while reading the media volume on device !	A tape I/O request failed. For more information, check the MPE status message printed after this message. The tape drive may need cleaning or repair, or the volume may be corrupt. Restart the restore with a different volume of the backup or use **recover media**.
An error occurred while reading virtual tape file !	The MPE FREAD intrinsic failed. For more information, check the file system error message printed after this message.
An error occurred while renaming virtual tape file !	A file system error occurred while renaming the virtual tape file. For more information, check the file system error message printed after this message.
An error occurred while rewinding media volume on device !	A tape I/O request failed. For more information, check the MPE status message printed after this message. The tape drive may need cleaning or repair, or the volume may be corrupt. Restart the restore with a different volume of the backup or use **recover media**.
An error occurred while routing console output to !	An invalid session number or user.account was specified as the target of the **display** keyword. It is possible that either the session or user.account specified has logged off since the operation began.

Error or Warning	Additional Information
An error occurred while scanning the tape on device !	A tape I/O request failed. For more information, check the MPE status message printed after this message. The tape drive may need cleaning or repair, or the volume may be corrupt. Restart the restore with a different volume of the backup or use **recover media**.
An error occurred while trying to find file ! to be purged	An MPE directory procedure failed while attempting to purge an existing file before the new file could be created. For more information, check the MPE status message printed after this message.
An error occurred while trying to obtain a pointer to a file label	The MPE procedure used to access this file's internal label failed. This often occurs when some volumes of a mountable volume set are not mounted. For more information, check the MPE status message printed after the error. The store of the affected file is cancelled.
An error occurred while trying to obtain current logon information	An MPE procedure failed. Please note the error message printed after this message and call your RoadRunner support engineer.
An error occurred while trying to read the ACD for file !	An MPE procedure failed while attempting to read the access control definition in the label extension for the file. For more information, check the MPE status message printed after this message.
An error occurred while updating the backup date file !	The MPE `FWRITEDIR` intrinsic failed. For more information, check the file system error message printed after this message.
An error occurred while writing to virtual tape file !	The MPE `FWRITE` intrinsic failed. For more information, check the file system error message printed after this message.
An I/O error occurred while reading logged data for file !	A error occurred while attempting to read logged information from the online log file. For more information, check the MPE status message printed after this message.
An internal printing error occurred, info: !	An unexpected error occurred while printing. Notify your RoadRunner support engineer.
An internal error occurred while trying to determine the number of disc devices on this system	RoadRunner is unable to determine the number of disc drives configured on this system. Without this information, RoadRunner is unable to operate properly. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
An invalid compression descriptor was found in a media block	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
An invalid compression type was found in a media block	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
An invalid data identifier was found in a media block	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.

Error or Warning	Additional Information
An invalid descriptor length was found in a media block	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
An invalid directory block was read from the media	There is data corruption in a directory of the backup. Use **recover media** to recover as many of the files as possible.
An invalid directory entry was found in a directory block	There is data corruption in a media block of the backup. Use **recover media** to recover as many of the files as possible.
An invalid file identifier was found in a media block	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
An invalid fill descriptor was found in a media block	Either the backup has been corrupted, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
An invalid linkage entry was found in virtual tape file !	The virtual tape file is corrupt. The files on this backup cannot be restored, validated or listed. The file may have been inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
An irrecoverable error occurred on device !	A tape I/O request failed. For more information, check the MPE status message printed after this message. If the operation was a store, the volume mounted should be flagged as bad, the tape drive cleaned and a new media volume mounted on the device. If this is an appended backup, additional cleanup is performed by RoadRunner to preserve the integrity of previous backups on the tape.
An unexpected continuation descriptor was encountered in a media block	There is data corruption in a media block of the backup. Use **recover files** to recover as much of the affected files as possible.
An unexpected continuation header was encountered	There is data corruption in a media block of the backup. Use **recover files** to recover as much of the affected files as possible.
An unexpected directory I/O error occurred	An error occurred while trying to read or write to the RoadRunner directory file. For more information, check the MPE status message printed after this message.
An unexpected end of data was encountered during decompression	The end of the backup media was encountered decompressing a section of data. This can sometimes happen when restoring files from an incomplete or aborted backup.
An unexpected error occurred while releasing an IPC message frame	The MPE procedure used to release internal messages failed. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
An unexpected error occurred while sending an IPC message	The MPE procedure used to send messages between RoadRunner processes failed. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.

Error or Warning	Additional Information
An unexpected error occurred while trying to set autoreply for device !	Please notify your RoadRunner support engineer.
An unexpected error occurred while updating the MPE/iX directory	The MPE FUPDATE intrinsic failed. For more information, check the file system error message printed after this message.
An unexpected I/O error occurred while accessing virtual tape file !	An MPE procedure failed while trying to read the directory of a virtual tape file. For more information, check the MPE status message printed after this message.
AUTOLOAD failed because device ! is already loaded	RoadRunner attempted to load a device that was already loaded and online. No action is necessary.
Backup ! does not exist on backup mounted on device !	The backup specified with the **index** keyword is not on the mounted backup. Verify that the **index** number is correct, and that the correct reel is mounted. If so, retry the operation with **recover media**.
Backup ! not found on reel ! of backup mounted on device !	The backup specified with the **index** keyword is not on the mounted backup. Verify that the number is correct, and that the correct reel is mounted. If so, retry the operation with **recover media**.
Backup date file! does not exist	The date file specified does not exist on the system. Verify that the name was typed correctly.
Backup id in backup header of file ! doesn't match backup id in directory	The identification number which uniquely identifies each backup created by RoadRunner differs between the data blocks and the directory blocks. The file may have been inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
Can't rename across group boundaries unless entire virtual tape fileset resides in one group	If a virtual tape fileset resides in more than one group, the name can be changed, but it cannot be relocated to different groups without destroying internal linkages.
Can't set autoreply for device !, autoreply is only valid for tape devices	The **autoreply** keyword was used, but the device specified in the **to** or **from** keyword was not a tape device.
Can't set autoreply for device !, it is being serviced by AVR	An attempt was made to set **autoreply** for a device which was being serviced by MPE's automatic volume recognition subsystem. RoadRunner attempted to wait for the operation to complete, but the device was still busy so **autoreply** was not set for the device.
Can't set autoreply for device !, it is not at load point	To successfully set **autoreply** for a device, it must either have a tape mounted and be at the load point, or no tape mounted at all. Check the tape drive and make sure that this is the case.
Cannot append backup ! to the backup on device ! because the highest backup in the set is !	If you wish to append to the end of a backup set, use **append** without specifying a backup index.

Error or Warning	Additional Information
Cannot append to backup on device !	An appended backup was attempted to a volume which is either not a RoadRunner backup, or not the last volume in a backup set. It is also possible that the previous backup was aborted. This volume cannot be used for appended backups.
Cannot close media path for file !	The FCLOSE intrinsic failed. For more information, check the file system error message printed after this message.
Cannot create the online environment	This error is often caused by lack of disc space. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
Cannot create include file	For more information, check the file system error message printed after this message.
Cannot create temporary file used in processing indirect file	For more information, check the file system error message printed after this message
Cannot find media trailer on device !	The appended backup specified with the **index** keyword could not be located. Either the backup is not on the mounted media volume, the volume was overwritten by another backup, or the media is corrupt. If you are certain the mounted volume is correct, and the backup with the specified index is on the tape, retry the command with the **recover media** option.
Cannot load online procedures	Unable to load the online logging procedures. This may happen if an online backup is attempted on an MPE/iX release prior to 3.0.
Cannot open indirectfile !	Make sure the file exists and was specified properly. For more information, check the file system error message printed after this message.
Cannot open media path for file !	An MPE file system intrinsic failed while trying to establish access to the backup device. For more information, see the error message printed after this message.
Cannot open report destination file !	An MPE file system intrinsic failed. For more information, see the error message printed after this message.
Cannot read indirectfile !	For more information, check the file system error message printed after this message.
Cannot retrieve the user interface language id, NLS error !	The user interface language is determined by a JCW value and is used to adjust various formats based on the native language desired. This message indicates that the user language cannot be retrieved, and 0 will be used (English).
Cannot set density to ! on device !	The density specified in the **to** or **from** keyword, or in the file equation used to access the storage device is not applicable for the drive being used, or for the tape being accessed. Verify that the density specified is compatible with your storage device and media.
Cannot set logical record pointer for indirect file	An MPE procedure failed. Please note the error message printed after this message and call your RoadRunner support engineer.
Cannot set logical record pointer for file used in processing indirect file	An MPE procedure failed. Please note the error message printed after this message and call your RoadRunner support engineer.

Error or Warning	Additional Information
Could only allocate a spool buffer of ! sectors	The **buffer** keyword was specified, but there was insufficient disc space to allocate one of the preferred size. A smaller buffer of the indicated size will be used.
database ! is not in a consistent state because file ! was not restored properly	During a restore with a **dbrestore** in effect, RoadRunner was unable to restore one or more of a database's data sets. The database is in an inconsistent state and should either be purged or completely restored from another backup.
database ! was not stored because one or more data sets were inaccessible	During a store with the **dbstore** option in effect, RoadRunner was unable to store one or more of a database's data sets. The backup of the remainder of the database was cancelled.
Data is missing for file !	When a restore is performed with the **single** keyword in effect, all files from one volume are restored, but subsequent volumes are not read. The partially restored file which was being restored at the end of the volume was not saved because the **recover files** keyword was not used.
Directory block sequence error on volume ! of volume set !	There are missing directory blocks on the media volume. The storage device may need cleaning or repair, or the media volume may be in poor condition. Verify that the drive is functioning correctly, then specify **recover all** to restore as much data as possible from the tape.
Directory block sequence error on volume !	There are missing directory blocks on the media volume. The storage device may need cleaning or repair, or the media volume may be in poor condition. Verify that the drive is functioning correctly, then specify **recover all** to restore as much data as possible from the tape.
Due to an error appending backup number !, backup ! has an incorrect media trailer. If you mount volume ! and attempt to restore backup !, errors will occur. Backup ! will begin on the next volume.	While writing an appended backup to a volume containing other backups, RoadRunner encountered a hard tape error. At this time RoadRunner tries to find the end of the last complete backup on the tape, and set a flag indicating that the backup continues on the next volume. This message occurs when this process fails. If you attempt to restore files from the backup that was being appended on this volume, RoadRunner will be unable to restore data beyond the point at which the error occurred.
Encountered end of data on $STDLIST	The command interpreter encountered an end-of-file condition on the standard input file. This usually happens in batch jobs by forgetting to add a RoadRunner exit command before the EOJ.
Error ! routing DISPLAY message to file !	An error occurred while routing a messages to the specified file. More information can be found by consulting the MPE/XL intrinsics manual.
Error occurred while writing to include file	For more information, check the file system error message printed after this message.
Error positioning tape for tape label. The density specified may exceed the maximum allowed.	The MPE HPFOPEN intrinsic failed indicating that there was a problem finding an ANSI label on the specified device. It is possible that an incorrect density has been specified.
Error writing directory to tape	An MPE procedure failed while writing to the backup media. For more information, check the MPE status message printed after this message.

Error or Warning	Additional Information
Fields may not be specified in the **report** keyword when using the **listinfo** command	The content of the **listinfo** report is pre-defined, so you cannot specify fields or `with` options. The **report** keyword is used to specify only the destination(s) and format of the report.
File ! could not be opened because it does not exist	An attempt was made to open a non-existent database root file for exclusive access in order to prevent user access during the restore of the database. This only occurs in conjunction with the **dbrestore** keyword.
File ! could not be opened exclusively	An attempt to open the database root file for exclusive access failed. The root file is opened exclusively to prevent access during the restore of the database. This only occurs in conjunction with the **dbrestore** keyword and is usually caused by user access to the database during the restore operation.
File ! not restored, an error occurred while creating the account	An MPE procedure failed trying to create a non-existent account. This error can occur either with the **directory** keyword, or in conjunction with the **create** option on restore. For more information, check the MPE status message printed after this message.
File ! not restored, an error occurred while creating the creator	An MPE procedure failed trying to create a non-existent user. This error can occur either with the **directory** keyword, or in conjunction with the **create** option on restore. For more information, check the MPE status message printed after this message.
File ! not restored, an error occurred while creating the group	An MPE procedure failed trying to create a non-existent group. This error can occur either with the **directory** keyword, or in conjunction with the **create** option on restore. For more information, check the MPE status message printed after this message.
File ! not restored, insufficient capabilities to create account	You must have SM or OP capability to use the create account option on restore.
File ! not restored, insufficient capabilities to create the creator	You must have SM, AM or OP capability to use the **create creator** option on restore.
File ! not restored, insufficient capabilities to create the group	You must have either SM, AM or OP capability to use the **create group** option on restore.
File ! not restored, non-existent account	Create the account or use the **create account** keyword.
File ! not restored, non-existent group	Create the group or use the **create group** keyword.
File ! purged BEFORE restore due to insufficient disc space	When there is insufficient disc space available in the volume set to restore a file, RoadRunner purges the existing file and attempts to allocate the space again. This warning indicates that the file was purged to provide space for the allocation of the new file. Check the restore listing to make sure that the file was restored successfully, and if not, free up additional disc space and retry the operation.

Error or Warning	Additional Information
File ! restored, but unable to save old dates in file label	An MPE procedure failed while trying to save the old creation, modification and access dates for a file. The file is restored, but these dates will reflect the date of the restore, and not original dates that were in place when the file was stored. For more information, check the MPE status message printed after this message.
Hardware timeout detected on device !	This can occur when the device is powered off, or when it is disconnected from the system. If this happens during the first access to the storage device, a request is issued to the console for confirmation to continue. If this occurs after a RoadRunner operation has begun, normal error handling is invoked, and the tape is rejected as part of the backup.
High compression was not purchased, default used	The RoadRunner Pro module has not been purchased, but `compress=2` was specified in the command.
Incorrect ! media volume (volume ! of set !) mounted on device !	Either mount the correct volume or use **recover media** to restore files from this volume.
Incorrect ! media volume (volume !) mounted on device !	Either mount the correct volume or use **recover media** to restore files from this volume.
Internal error printing message, info: !	A RoadRunner message could not be found or printed. Call your RoadRunner support engineer.
Invalid ACCOUNT name/pattern for '!' in indirectfile !	The account name you specified doesn't meet MPE file naming conventions.
Invalid combination of path specifications	The device paths specified for the current command are not compatible. Combinations of disc, tape and/or DAT are invalid.
Invalid device class specification: !	A device class was specified in the command or file equation which was neither a tape nor a disc.
Invalid device specification: !	A device number was specified in the command or file equation which was neither a tape nor a disc.
Invalid device specified in file equation for directory	A file equation was entered for `RRXLDIR.PUB.SYS`, followed by a device specification (`DEV=`) which was invalid. If a volume or volume class was specified in the file equation, make sure it is a valid member of the home volume set of the group to which RoadRunner directory file was routed.
Invalid disc specification: !	The disc device number specified in the command or file equation is either not a disc or the disc is not currently mounted on the system.
Invalid FILE name/pattern for '!' in indirectfile !	The file name you specified doesn't meet MPE file naming conventions.
Invalid file header encountered during decompression	Either the backup is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
Invalid file overlap	There is data corruption in a media block of the backup. Use **recover files** to recover as much of the affected files as possible.

Error or Warning	Additional Information
Invalid GROUP name/pattern for '!' in indirectfile !	The group name you specified doesn't meet MPE file naming conventions.
Invalid response type, expected !	A user or operator response did not match the data type expected. The type of response expected is indicated in the message and can be either `yes / no`, `character` (alphanumeric) or `integer` (numeric).
Invalid volume set number found on device !, use **recover media**	A volume set number greater than 16 was found on the media. This is possibly due to data corruption, and the tape in use may be of questionable quality. To restore from this backup, specify **recover media**.
Maximum number of soft tape errors exceeded: !	The **maxerrs** keyword was specified, and the soft error threshold was exceeded for the current volume. The tape drive may need cleaning or repair, or the media volume itself may be in poor condition. The volume which was mounted will not be a part of the current backup and a new volume should be mounted.
Media block sequence error on volume ! of volume set !	There are missing data blocks on the media volume. The storage device may need cleaning or repair, or the media volume may be in poor condition. Verify that the drive is functioning correctly, then specify **recover all** to restore as much data as possible from the tape.
Media block sequence error on volume !	There are missing data blocks on the media volume. The storage device may need cleaning or repair, or the media volume may be in poor condition. Verify that the drive is functioning correctly, then specify **recover all** to restore as much data as possible from the tape.
Media on device ! not recognized as RoadRunner backup	The media mounted on the specified device is not recognized as a RoadRunner backup. You may have mounted a hardware-compressed tape on a drive that doesn't support compression. This error can also be caused by media corruption, the overwriting of the first part of the tape, or an operator error in mounting the tape. Verify that the correct tape is mounted, or specify **recover media** to restore files from this backup.
Media set ! has already been processed	A volume was mounted for a read operation that is a member of a media set that has already been processed. Verify that the requested tape was mounted correctly. If the correct volume was mounted, it is possible that the media is corrupt. If this is the case, abort the operation and retry with **recover media**.
Media set ! is currently mounted on another device path	A tape was mounted for a read operation that is a member of a media set that is being read by another path. Verify that the requested tape was mounted correctly. If the correct tape was mounted, it is possible that the media is corrupt. If this is the case, abort the operation and retry with **recover media**.
MPE command failure, error !: !	The `HPCICOMMAND` intrinsic failed while executing an MPE command during event/action processing. Note the error number and consult the MPE/iX Commands Reference manual if necessary to correct the situation.
MPE/iX status: Subsystem !, error !	Indicates the error status returned from an MPE procedure such as `HPFOPEN`. It is preceded by a standard RoadRunner error message indicating the nature of the problem encountered. In some cases, RoadRunner aborts the current operation; in others, the affected file is simply not processed.

Error or Warning	Additional Information
No device specification found in file equation for designator '!'	The file equation for the file indicated did not have a device specification (dev=). Check the file equation and re-enter if necessary.
No file equation found for designator '!'	Make sure that you entered a file equation for the file indicated and that the name is spelled correctly. The :LISTEQ command can be used to get a list of current file equations.
No files were found to copy	The files requested were not found on the backup. Check spelling and selection criteria.
No files were found to restore	The files requested were not found on the backup. Check spelling and selection criteria.
No files were found to validate	The files requested were not found on the backup. Check spelling and selection criteria.
No media trailer was found at the end of the volume on device !	RoadRunner writes a trailer at the end of each volume to indicate the end of the data on a reel. If you aborted the operation or had a hardware error at the end of a reel, this trailer may not be written. Without it, there is no clear end to the data and no way of knowing if data has been lost. Use the **recover media** option to restore as many files as possible.
No MPE/iX directories stored on this backup	A restore was invoked with either the **directory** keyword or the **create usedir** keywords. The backup header indicates that the backup was not stored with the **directory** keyword and thus no directories are accessible.
Non-existent device class specified: !	The device class specified in the command or file equation is not configured on the system.
Non-existent device specified: !	The device number specified in the command or file equation is not configured on the system.
Nonexistent virtual tape file !	Check the spelling of the virtual tape filename specified in the command.
Only minimum directory information stored on this backup	The **directory minimum** keyword was specified when the backup was created. Therefore, only directories present on the backup will be restored.
Out of disc space in virtual tape fileset !	The operator has replied that there is no disc space left for a new virtual tape file to be created. The file that was being created is deleted, and the operation aborts.
Overwrite of mismatched internal label not confirmed	The RoadRunner label on the backup does not match the **volid** specified with the **to** keyword. The operator replied no to the request to overwrite the backup.
Overwrite of unexpired backup not confirmed	An attempt was made to overwrite a backup with an unexpired RoadRunner or ANSI label. The operator replied no to the request to overwrite the backup.
Parallel media sets is a feature which was not purchased	The RoadRunner PRO module has not been purchased, but parallel paths were specified in the command.

Error or Warning	Additional Information
Physical end-of-file reached while writing to include file	To contain all the information in the indirectfile being translated, the include file would be larger than the file system permits. Split your indirectfile into two or more files.
Purge of existing virtual fileset not confirmed, function aborted	The user did not confirm the deletion of an existing virtual tape file during a **store** or **copy** operation. The operation is aborted.
Recover incorrect volume on device ! (Y/N) ?	A volume other than the one expected by RoadRunner was mounted for reading and **recover media** was specified. If this is the volume you wish to read, reply 'Y', otherwise reply 'N' and RoadRunner will wait for you to mount the another volume.
Recover volume without directory on device ! (Y/N)?	A volume with a corrupt directory has been mounted and **recover media** was specified. If this is the volume you wish to read, reply 'Y'. Otherwise, reply 'N' and RoadRunner will wait for you to mount the correct volume. If a restore is started with this volume, a prompt will be issued at the end of each reel to determine if you wish to continue the restore.
Remote restore operations from a DAT unit cannot utilize RoadRunner's fast search capability. Therefore, this restore may take considerably longer than if it were performed locally	RoadRunner cannot access the DAT drive's fast search mechanism when operating over a network. Although the restore should complete successfully, it could take much longer than usual to locate specific files.
Report from $NULL is not allowed	You must specify a **from** device path other than `$null` when listing files.
Requested density for device ! is not supported by this device	Change the density specified in either the **to** or **from** keyword, or the den parameter of the file equation to one that is supported by the device.
Restore from $NULL is not allowed	You must specify a **from** device path other than `$null` when restoring files.
Restore of file ! cancelled due to backup media error	The file specified was not restored because an error occurred while reading the backup medium. The **recover files** option can be used to request that all restorable data be saved for all files being restored, regardless of media errors.
RoadRunner aborted due to media write error	An irrecoverable I/O error has occurred and **after error quit** was specified in the command.
Root file names for virtual tape files do not match	When doing a RoadRunner virtual tape store or restore using multiple paths, the "root" file name specified for each path must be the same. Check the file names of all paths to make sure they match.
Some files were not complete (SINGLE in effect)	When a restore is performed with the **single** keyword in effect, all files from one volume are restored, but subsequent volumes are not read. Files partially restored at the end of the volume are not saved unless the **recover files** keyword is used. The number of files affected depends on the interleave level used when the files were backed up.

Error or Warning	Additional Information
The **append** keyword cannot be used on remote devices	The **append** keyword can only be used with local DDS devices. Either perform the backup to a local device or remove the append keyword from the command and backup to a different cartridge.
The **append** keyword can only be used on DDS devices	The **append** keyword can only be used with local DDS devices. Either perform the backup to a DDS device, or remove the **append** keyword from the command and backup to a different cartridge.
The **autoeject** keyword can only be used on local DDS devices	The **autoeject** keyword was specified, but cannot be executed because the device being accessed is not recognized as a DDS device or is on a remote system.
The **autoeject** operation failed on device !	An error was encountered while attempting to eject a cartridge from a DDS device. For more information, check the MPE status message printed after this message.
The **autoload** feature can only be used when a device number is specified	The **autoload** keyword was specified in conjunction with a path for which a device class or remote device was specified. Specify the device number of the drive(s) with a file equation or the **to** keyword. The **autoload** feature is not supported with remote devices.
The **autoload** feature failed on device !	The MPE HPDEVCONTROL intrinsic failed. For more information, check the MPE status message printed after this message.
The **autoload** feature is not supported for device !	The tape must be loaded manually.
The **autoload** keyword was specified in conjunction with a path for which a file equation or device class was specified. The **autoload** function only supports device numbers.	Specify the device number of the drive(s) with a file equation or the **to** keyword.
The **autoreply** feature can only be used when a device number is specified	The **autoreply** keyword was specified in conjunction with a path for which a device class or remote device was specified. Specify the device number of the drive(s) in a file equation or the **to** keyword. The **autoreply** feature is not supported with remote devices.
The backup ! with volume id (!) has not yet expired	An attempt was made to overwrite a backup with an unexpired RoadRunner or ANSI label. To overwrite the volume, either specify the **ignlabel** keyword or reply 'Y' to the prompt asking if you wish to overwrite the existing backup.
The backup file ! has no user label and thus does not match the user specified id	A RoadRunner label was specified in the **from** keyword, but the virtual tape backup being accessed does not have a RoadRunner label. Either specify the correct backup, or remove the **label** portion of the **from** keyword and retry the operation.
The backup header is corrupt	The backup header is unidentifiable. This may be due to media corruption, the overwriting of the first part of the tape, or an operator error in mounting the tape. To restore from this backup, specify **recover media**.

Error or Warning	Additional Information
The backup media volume mounted on device ! has no internal user label and thus does not match the user specified id	A RoadRunner label was specified in the **from** keyword, but the backup volume being accessed does not have a RoadRunner label. Either mount the correct volume, or remove the **label** portion of the **from** keyword and retry the operation.
The density of all devices in a serial pool must be the same	There are different densities being used for devices within a serial device pool. Use the **density** parameter of the **to** or **from** keyword, or the **den** parameter of a file equation to set the density of all devices in the pool to the same density.
The fileset '!' in indirectfile ! contains invalid characters	The fileset specified doesn't meet MPE file naming wild card conventions.
The identification block in VT file ! is invalid	Either the file is corrupt, or the version of RoadRunner which created the backup is newer than the version being used to access it. Check to ensure that you are using the correct version of RoadRunner. If you are, use **recover files** to recover as much of the affected files as possible.
The **index** keyword cannot be used on remote devices	The **index** keyword can only be used on local DDS devices. To restore an appended backup, use a device connected to the local system.
The **index** keyword can only be used on DDS devices	The tape you are trying to access is not recognized as a DDS cartridge, and therefore cannot contain appended backups. Remove the **index** keyword from the command.
The labeled tape feature was not purchased	The RoadRunner PRO module has not been purchased, but ANSI labels were specified in the command.
The media on device ! is bad	A hard I/O error was encountered while writing to the storage device. The volume mounted should be flagged as bad and a new media volume mounted on the device.
The media volume on device ! is not a member of the current storeset	Either mount the correct volume or use **recover media** to restore files from this volume.
The MPE/iX directory open failed	The MPE internal directory open procedure failed. Please contact your RoadRunner support engineer.
The MPE/iX directory scan failed	The MPE internal directory scan procedure failed. Please contact your RoadRunner support engineer.
The number of **from** and **to** paths must be the same	For the **copy** command, the number of parallel device paths you are copying from must match the number of paths you are copying to. This is true regardless of the setting of the **combine** keyword or whether or not serial device pools are used.
The **online** feature was not purchased	The RoadRunner Online Module has not been purchased, but **online** was specified in the command.
There are more device paths specified than files	The number of parallel device paths must be less than or equal to the number of files to be backed up. Retry the command with fewer parallel device paths.

Error or Warning	Additional Information
There are no candidates to process on volume set !	A tape was mounted for a read operation that is a member of a media set on which there are no candidates to be processed. Verify that the requested tape was mounted correctly. If the correct tape was mounted, it is possible that the media is corrupt. If this is the case, abort the operation and retry with **recover media**.
The RoadRunner directory cannot be located on the media volume	This may be because the mounted volume is not a RoadRunner tape, or the media is corrupt. If the volume mounted is correct, use **recover media** to read from this volume.
The volume id '!' on backup ! does not match the user specified id	The RoadRunner label on the backup being accessed does not match the userid specified with the **from** keyword. Either specify the correct backup, or remove the **label** portion of the **from** keyword and retry the operation.
The volume id '!' on device does not match the user specified id	The RoadRunner label on the backup volume being accessed does not match the userid specified with the **from** keyword. Either mount the correct backup, or remove the **label** portion of the **from** keyword and retry the operation.
The volume id specified on the internal label does not match	The RoadRunner label on the backup does not match the userid specified with the **from** keyword and no other backup was available. To restore from this backup, change the userid with the **from** keyword to match the userid of the backup from which you wish to restore or remove the **label** portion of the **from** keyword and retry the operation.
The volume restriction for file ! was changed to volume class DISC due to insufficient disc space in the file's original restriction	The volume restrictions that were to be applied while restoring the specified file would place it on a volume or in a volume class that did not have enough disc space to hold the file. These volume restrictions come from any specified **vol**, **volclass**, or **volset** keywords that may have been specified, or from the file label if none were specified. To find out the volume restrictions of a file, perform a **listdir** on the backup and specify the **volinfo** parameter of the **report** keyword.
The volume restriction for file ! was changed to volume set ! due to insufficient disc space in the file's original specification	The volume restrictions that were to be applied while restoring the specified file would place it on a volume or in a volume class that did not have enough disc space to hold the file. These volume restrictions come from any specified **vol**, **volclass**, or **volset** keywords that may have been specified, or from the file label if none were specified. To find out the volume restrictions of a file, perform a **listdir** on the backup and specify the **volinfo** parameter of the **report** keyword.
The **with** section of the **report** keyword may not be specified when using the **listinfo** command	The content of the **listinfo** report is pre-defined, so you cannot specify `with` options. The **report** keyword is used to specify only the destination(s) and format of the report.
This backup has an invalid data format (!). Use a newer version of RoadRunner to read from this volume	The format of this backup is incompatible with the version of RoadRunner you are running. Use MPE `STORE/RESTORE` to restore `ROADREST.PUB.SYS` from the backup, then use `ROADREST` to perform the operation.
Too many devices are configured on this system. The maximum supported is 5120	The internal RoadRunner tables are configured for a maximum of 5120 configured devices. Should this error occur, please contact your RoadRunner support engineer immediately.

Error or Warning	Additional Information
Unable to allocate an object control block	Out of available heap space to allocate an internal data structure. Please notify your RoadRunner support engineer.
Unable to allocate disc space for an internal buffer	The MPE procedure used to allocate disc space for RoadRunner's internal buffer file failed. For more information, check the MPE status message printed after this message.
Unable to allocate disc space for file !	An MPE procedure failed while attempting to allocate disc space for the file. For more information, check the MPE status message printed after this message.
Unable to allocate space for the directory	It is possible that there is not enough free disc space to complete the build. For more information, check the MPE status message printed after this message.
Unable to apply volume restrictions for file !	When restoring the specified file, the volume restrictions that were to be applied could not be enforced. The file is still restored, but it is simply placed on the volume set to which its group is bound. If the **vol**, **volclass**, or **volset** keywords were used, the entity specified by these keywords either does not exist on the system, or is not mounted. Check the spelling of any volumes, volume classes, or volume sets in the command and ensure that these entities exist and are mounted. If none of these keywords was used, the volume restrictions that were in the file label are used at restore time. In this event, check to see that the volume, volume class, or volume set indicated by the file's label is configured and mounted. To determine the original volume restrictions of a file, perform a **listdir** and specify the **volinfo** parameter with the **report** keyword. To restore files with volume restrictions other than those specified in the file label, use the **vol**, **volclass**, or **volset** keyword.
Unable to attach ! to the online subsystem	An unexpected error occurred while establishing linkages to the online logging subsystem. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
Unable to close report file !, report renamed to !	A disc file was used as a report destination, but the filename used already exists on the system. When this occurs, RoadRunner automatically renames the new report to `rrhhmmss`, where `hhmmss` is the hour, minute and second at which the report was saved.
Unable to close store file !	The MPE procedure which releases access to the file failed, but the file was successfully backed up. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to close virtual tape file !	The MPE FCLOSE intrinsic failed. For more information, check the file system error message printed after this message.
Unable to create a child process	The MPE CREATEPROCESS intrinsic failed while trying to create a RoadRunner child process. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to create an internal object	The MPE HPFOPEN intrinsic failed while trying to create an internal file object. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to create buffer file !	The MPE HPFOPEN intrinsic failed while trying to create the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.

Error or Warning	Additional Information
Unable to create file ! to be restored	The MPE HPFOPEN intrinsic failed while trying to create the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to create message pool	The MPE procedure which creates an interprocess communications message pool failed. For more information, check the MPE status message printed after this message.
Unable to create MPE/iX directory backup file !	The MPE HPFOPEN intrinsic failed while trying to create the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to create the backup date file !	The MPE HPFOPEN intrinsic failed while trying to create the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to create the directory file	The MPE HPFOPEN intrinsic failed while trying to create the RoadRunner directory file. For more information, check the MPE status message printed after this message.
Unable to create the extent map for file !	The file was not stored because the MPE procedure which provides extent allocation information failed. For more information, check the MPE status message printed after this message.
Unable to create virtual tape file !	The MPE HPFOPEN intrinsic failed while trying to create the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to determine the current version of MPE	The method by which RoadRunner determines the version of the operating system on which it is running failed. Notify your RoadRunner support engineer.
Unable to find the file directory in virtual tape file !	The virtual tape file is corrupt. The files on this backup cannot be restored, validated or listed. This is possibly because the file was inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
Unable to find the identification block in virtual tape file !	The virtual tape file is corrupt. The files on this backup cannot be restored, validated or listed. This is possibly because the file was inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
Unable to find volume label in virtual tape file !	The virtual tape file is corrupt. The files on this backup cannot be restored, validated or listed. This is probably because the file was inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
Unable to freeze a buffer in memory	The MPE procedure used to lock RoadRunner's internal buffer in memory failed. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to get a USER directory entry for !.!	RoadRunner was unable to find a user entry for the creator of a file being stored to be added to the MPE/iX directory backup. This usually occurs when the creator of the file has been purged with a :PURGEUSER command.

Error or Warning	Additional Information
Unable to get file creation information for file !	The MPE FFILEINFO intrinsic failed while trying to determine the creator of a virtual tape backup file. For more information, check the file system error message printed after this message.
Unable to get file information for the directory file	The FFILEINFO intrinsic failed. For more information, check the file system error message printed after this message.
Unable to get group home volume set information for group !	An MPE procedure failed while trying to obtain the group's home volume set name. This error can occur either when RoadRunner is trying to change a file's volume restrictions due to insufficient disc space or when the **volset @** keyword is used. For more information, check the MPE status message printed after this message.
Unable to get information about an internal object	The MPE FFILEINFO intrinsic failed. For more information, check the file system error message printed after this message.
Unable to get volume restrictions for file !	An MPE procedure failed while RoadRunner was attempting to determine the volume restrictions for a file to be restored. Check to ensure that the specifications used with the **vol**, **volclass**, or **volset** keywords are correct and that all volumes are currently mounted.
Unable to initialize the backup date file !	The MPE FWRITEDIR intrinsic failed. For more information, check the file system error message printed after this message.
Unable to locate file for volume ! of volume set ! of virtual tape fileset !	A file that is a member of the virtual tape fileset cannot be found. This is possibly because the file was somehow deleted from the system, or the store process that created the virtual tape fileset was aborted. Files that were stored to this or subsequent virtual tape files in the fileset are inaccessible.
Unable to locate linkages in virtual tape file !	The virtual tape file is corrupt. The files on this backup cannot be restored, validated or listed. This is probably because the file was inadvertently written to by software other than RoadRunner, or the store process that created the virtual tape fileset was aborted.
Unable to locate the MPE/iX directory backup	A restore was invoked with the **directory** keyword; the backup header indicated that the backup indeed was created with the **directory** keyword, but the MPE directory backup file could not be opened.
Unable to map in file !	RoadRunner was unable to map the specified file into virtual memory. For more information, check the MPE status message printed after this message.
Unable to obtain an IPC message port	The MPE procedure that creates an interprocess communications message port failed. For more information, check the MPE status message printed after this information about the problem.
Unable to obtain an IPC message	The MPE procedure used to obtain interprocess messages failed. Please note the MPE status message printed after this message and contact your RoadRunner support engineer.
Unable to obtain file information about XL directory file !	The the MPE FFILEINFO intrinsic failed. For more information, check the file system error message printed after this message.
Unable to obtain online logging information for file !	An error occurred while requesting information for a file being backed up using the **online** keyword. For more information, check the MPE status message printed after this message.

Error or Warning	Additional Information
Unable to obtain the JDT address	RoadRunner was unable to locate the job directory table. This table is used to determine what file equations, if any, are in effect. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to obtain virtual tape file information for file !	The the MPE FFILEINFO intrinsic failed. For more information, check the file system error message printed after this message.
Unable to open file ! to be stored	RoadRunner was unable to open the specified file. The backup of the file was cancelled. For more information, check the MPE status message printed after this message.
Unable to open RoadRunner help file: !	The RoadRunner help file could not be opened. The RoadRunner help file is called RRHLP000, and should reside in the same group and account as the program.
Unable to open the backup date file !	After the successful creation of the backup date file, the MPE HPFOPEN intrinsic failed to re-open the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to open virtual tape file !	The MPE HPFOPEN intrinsic failed while trying to open the file. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to open XL directory backup file !	The MPE/iX directory backup file was successfully created as a temporary file, but the MPE HPFOPEN intrinsic failed during an attempt to re-open it. For more information, check the error number printed after this message in the MPE/iX intrinsics manual.
Unable to post buffer data to disc	The MPE procedure which posts data to the internal buffer file failed. For more information, check the MPE status message printed after this message.
Unable to purge existing XL directory backup file !	The MPE FCLOSE intrinsic failed while trying to purge an old XL directory backup file. For more information, check the file system error message printed after this message.
Unable to purge file !	A request was made to purge the specified file after backup, but an error occurred when the purge was attempted. For more information note the error message printed after this message.
Unable to purge virtual tape file !	The MPE FCLOSE intrinsic failed. For more information, check the file system error message printed after this message.
Unable to read an IPC message	The MPE procedure which reads interprocess communications messages failed. For more information, check the MPE status message printed after this message.
Unable to read from an internal buffer	The MPE procedure used to read from RoadRunner's buffer file failed. For more information, check the MPE status message printed after this message.
Unable to read from XL directory backup file !	The MPE FREAD intrinsic failed. For more information, check the file system error message printed after this message.
Unable to read the backup header from virtual tape file !	The MPE FREAD intrinsic failed. For more information, check the file system error message printed after this message.

Error or Warning	Additional Information
Unable to read the file label extension for file !	An error occurred while RoadRunner was trying to read a file label extension, such as the file's ACD, spool file information or KSAM/XL internal information. For more information, check the MPE status message printed after this message.
Unable to release file ! from on-line subsystem	RoadRunner was unable to release a file from the online logging subsystem. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to rename !, System Manager capability needed to rename across accounts	The user has insufficient capability to rename files across account boundaries. Retry the `rename` with a user that has System Manager (SM) capability.
Unable to reset store bit for file !	The MPE procedure which resets store bits failed. For more information, check the MPE status message printed after this message.
Unable to save file ! as a permanent file	The MPE `FCLOSE` intrinsic failed. This could be due to either a directory problem or a problem encountered while trying to purge the existing file on disc before the restore. For more information, check the file system error message printed after this message.
Unable to save the backup date file ! as a permanent file	The MPE `FCLOSE` intrinsic failed. For more information, check the file system error message printed after this message.
Unable to save XL directory backup file ! as a temporary file	The MPE `FCLOSE` intrinsic failed. For more information, check the file system error message printed after this message.
Unable to set the store bit for file !	This error indicates that a problem occurred while setting the store bit for a file; if the file was open for writing or the store bit was already set, a different message is printed. For more information, check the MPE status message printed after this message.
Unable to unfreeze a buffer from memory	The MPE procedure used to release RoadRunner's internal buffer from memory failed. Please note the MPE status message printed and contact your RoadRunner support engineer.
Unable to write backup header to virtual tape file !	The MPE `FWRITE` intrinsic failed. For more information, check the file system error message printed after this message.
Unable to write to XL directory backup file !	The MPE `FWRITE` intrinsic failed. For more information, check the file system error message printed after this message.
Unexpected end of logged data for file !	More logged online transaction data was expected for the file indicated; however, the log index indicates that there is no more data left in the log for this file. Please notify your RoadRunner support engineer.
Validate from $NULL is not allowed	You must specify a **from** device path other than `$null` when validating files.
Validate of file ! cancelled due to backup media error	An error was encountered while reading a media block from tape. This could be a hardware or media problem and results in a corrupt media block. You may not be able to restore the file specified unless you use the **recover files** keyword, and in that case all data for the file will not be recovered.

Error or Warning	Additional Information
Virtual tape and tape cannot be used on the same path	A mixture of virtual tape files and device files was specified in either the **to** or the **from** keyword. Examine your command to ensure that all members of a serial device pool are either like devices or virtual tape files, not a mixture of both.
Volume ! is not mounted	The volume specified with the **volume** keyword is not currently mounted. Specify another volume or ensure that the requested volume is mounted.
Volume ! was not found in volume set !	The volume specified with the **volume** keyword is not configured as a member of the specified volume set. Check the volume set indicated or if no volume set was specified, specify the **volset** keyword followed by the appropriate volume set name.
Volume class ! is not defined for volume set !	The volume class specified with the **volclass** keyword is not configured for the specified volume set. Check the volume set indicated or if no volume set was specified, specify the **volset** keyword followed by the appropriate volume set name.
Volume number ! on device ! with volume id '!' has not yet expired	An attempt was made to overwrite a backup with an unexpired RoadRunner or ANSI label. To overwrite the volume, either specify the **ignlabel** keyword or reply 'Y' to the prompt asking if you wish to overwrite the existing backup.
Volume set ! does not exist or is not currently mounted	The volume set specified with the **volset** keyword does not exist on the system or is not currently mounted. Verify that the name was typed correctly.
Volume set ! was not found, will create group !.! in the system domain	When restoring the MPE/iX directory, RoadRunner attempted to create the specified group on a volume set that does not exist on the system. The group is built in the system domain (`MPEXL_SYSTEM_VOLUME_SET`). This can occur when an MPE/iX directory is restored on a system where a volume set name has been changed since the backup was performed, or when an MPE/iX directory is restored on a system other than the one on which it was created.

F

G

H

I

J

K

L

Q

R

S

UNISON ROADRUNNER FOR MPE

Corporate Headquarters
5101 Patrick Henry Drive
Santa Clara, CA 95054
408/988-2800
Fax 408/988-2236

Austin Division
811 Barton Springs Road
Austin, TX 78704
512/478-0611
Fax 512/479-0735

European Headquarters
Mardall House, Ground Floor
9-11 Vaughan Road, Harpenden
Hertfordshire AL5 4HU
United Kingdom
+44 (0) 1582 462424
Fax +44 (0)1582 462425

RR-900000-1196-0