Contents

1 Introduction to NetWare Btrieve

2 NetWare Btrieve Architecture

3 Installing and Configuring NetWare Btrieve

4 Configuring and Using the Requesters

5 Using NetWare Btrieve Utilities

A Btrieve Concepts

B Description Files

C Status Codes and Messages

Index

chapter **1**

Introduction to NetWare Btrieve

The Btrieve® key-indexed record management system is designed for high-performance data handling and improved programming productivity. Btrieve allows an application to retrieve, insert, update, or delete records either by key value, or by sequential or random access methods.

This chapter contains the following sections:

- "Client-Server Design" on page 1
- "What Is a Client?" on page 3
- "NetWare Btrieve Features" on page 4
- "Enhancements to NetWare Btrieve" on page 7

Client-Server Design

Btrieve products area available in the form of server-based Btrieve (also called NetWare® Btrieve™) and client-based Btrieve. In most cases, applications written for server-based Btrieve can run on client-based Btrieve and vice versa.

Server-based Btrieve, which is provided with the NetWare operating system, includes the following components:

Checklist ✓

□ Server-based Btrieve Record Manager (that is, the server-based Btrieve engine), which runs at the server as an NLM and manages data I/O with the file system.

□ Btrieve communications programs, which handle incoming requests from a remote source and which can route requests from a server-based application to a copy of the Record Manager running on a remote server.

❑ Btrieve Requesters, which run at the workstation and handle data I/O between the workstation and the server.

The Requesters allow applications running at the workstation to communicate transparently with the Record Manager on the server. Btrieve Requesters are available for DOS, OS/2*, and MS Windows* workstations.

❑ Btrieve utilities, which provide installation, setup, rebuild, monitor, maintenance, and recovery programs for Btrieve users and files.

❑ Novell Directory Services™ (NDS™) support utility, which allows the Record Manager to register with NDS.

❑ *NetWare Btrieve Installation and Operation* manual (this document), which describes using Btrieve in a NetWare environment.

Client-based Btrieve is available only as part of a Btrieve Developer’s Kit, which is available through Btrieve Technologies, Inc. at 1-800-BTRIEVE (for the DOS, OS/2, or MS Windows environment) and must be purchased separately. Each Btrieve Developer’s Kit includes the following components:

Checklist ✓

❑ Client-based Record Manager for the applicable environment (for use with the DOS, OS/2, or MS Windows operating systems).

The client-based Record Manager (that is, the client-based Btrieve engine) executes all processing on the workstation. It accesses Btrieve files through operating system calls, which are either executed locally (for files stored locally), or redirected to a server (for files stored on a server).

❑ Btrieve utilities, which provide installation, setup, rebuild (Btrieve v5.*x* to v6.*x* files), maintenance, and recovery programs for Btrieve users and files.

❑ *Btrieve [for the X Environment] Installation and Operation* manual, which documents using Btrieve in the applicable environment (the DOS, OS/2, or MS Windows operating systems).

❑ *Btrieve [for the X Environment] Programmer’s Manual*, which documents the Application Programming Interface (API) and the language interfaces that allow Btrieve to be called from various programming languages.

What Is a Client?

Throughout this manual and in other Btrieve documentation, you will see the term *client* used in several ways. The meaning of the term varies according to the context in which it appears.

As used in the NetWare client-server environment, the term *client* usually refers to a workstation that requests file or printing services from a NetWare file server. Parts of this manual (as well as the *Btrieve Programmer's Manual*) use the term in this manner.

Both manuals also use *client* to refer to the Btrieve engine running on a particular workstation. In this situation, client refers to a single piece of software running on the workstation (as opposed to the workstation itself).

In other parts of both manuals, *client* refers to each task currently running on a workstation. In a multitasking environment such as MS Windows, multiple clients can be active on the workstation at any given time. Therefore, when client is used in this manner, it usually identifies tasks that are instances of one or more Btrieve applications that are running simultaneously on a workstation.

Yet other parts of both manuals use the term *client* to refer to entities that a programmer can define and create in an application and then manipulate by using the **btrvid** or **btrcallid** functions. In this environment, multiple instances of an application can be running at the same time on a workstation, and each of those instances can have multiple Btrieve clients that it is manipulating.

As implied in the preceding paragraphs, when you see the term *client* in this manual (or in the *Btrieve Programmer's Manual*), remember that is meaning is determined by the context in which it appears.

Note The *Btrieve Programmer's Manual* is available only as part of the Btrieve Developer's Kit, which is available through Btrieve Technologies, Inc. at 1-800-BTRIEVE.

NetWare Btrieve Features

The following sections introduce some of the features that make the NetWare Btrieve record management system uniquely powerful.

Index Maintenance

NetWare Btrieve automatically creates and maintains the indexes in your files as you insert, update, and delete records. In addition to automatic index maintenance, NetWare Btrieve supports the following index features:

- Up to 119 key segments per file
- Adding or dropping any index after a file has been created
- Numerous data types for key values: `integer`, `float`, `date`, `time`, `decimal`, `money`, `logical`, `numeric`, `bfloat`, `string`, `lstring`, `zstring`, `unsigned binary`, `autoincrement`, and `signed trailing separate`
- Numerous key attributes: linked/repeating duplicatable, duplicatable/ nonduplicatable, supplemental, modifiable/nonmodifiable, segmented/ nonsegmented, descending/ascending sorting, case-sensitive/case-insensitive sorting, alternate collating sequence, null (any-segment/all-segment)/non-null

For more information, see “Indexes” on page 231 in Appendix A, “Btrieve Concepts.”

File Specifications

NetWare Btrieve allows you to create Btrieve (data) files by using either function calls (from an application) or the Btrieve Maintenance utility (BUTIL) commands. NetWare Btrieve offers these file specifications:

- File sizes up to 4 billion bytes (4 gigabytes)
- Number of records limited only by the size limit of the file
- Consistent file definition and management routines independent of the operating environment
- Consistent file structures

Memory Management

The cache is an area of memory that NetWare Btrieve reserves for buffering the pages that it reads. The cache is divided into a number of buffers, each of which is initially the size of the largest page the application will access. NetWare Btrieve calculates this size using the maximum page size and the memory options you define when you configure NetWare Btrieve.

Generally, a larger cache improves performance because it allows more pages to be in memory at a given time. NetWare Btrieve allows you to specify the amount of memory to reserve for the I/O cache buffers.

To determine this amount of memory, take into consideration the following factors:

- The application’s memory requirements (if the application is an NLM)
- The total amount of memory installed on the workstation (for client-based Btrieve) or on your server (for NetWare Btrieve)
- The combined size of all files the application will access

When your application requests a record, NetWare Btrieve first checks the cache to see if the page containing that record is already in memory. If so, NetWare Btrieve transfers the record from the cache to the application’s data buffer.

If the page is not in the cache, NetWare Btrieve reads the page from the disk into a cache buffer before transferring the requested record to the application.

If every cache buffer is full when NetWare Btrieve needs to transfer a new page into memory, a least-recently-used (LRU) algorithm determines which page in the cache Btrieve should overwrite. The LRU algorithm reduces processing time by keeping the most recently referenced pages in memory.

When an application inserts or updates a record, NetWare Btrieve first modifies the corresponding page in the cache and then writes that page to disk. The modified page remains in the cache until the LRU algorithm determines that Btrieve can overwrite the image of that page in cache with a new page.

Data Integrity

The following NetWare Btrieve features let you support concurrent access while ensuring the integrity of your files in a multiuser environment:

- Single-record and multiple-record locks
- Concurrent and exclusive transactions (See “Concurrent Transactions” on page 17.)
- Deadlock detection (in a server-based environment)
- Shadow paging, which entails making changes to a copy of a page rather than to the original page when inserting, updating, or deleting a record (See “Shadow Paging” on page 19.)
- Logging feature, which records (in a log file) any changes made to a designated file
- Roll Forward utility, which uses log files maintained by Btrieve’s logging feature to recover data corrupted by a system or server failure

Security Controls

NetWare Btrieve provides the following capabilities for enhancing data security in a network environment:

- Assigning owner names to files

- Specifying dynamic encryption and decryption of data
- Providing NetWare file-level security through the NetWare Btrieve Requesters

Enhancements to NetWare Btrieve

NetWare Btrieve 6.*x* includes many features and performance enhancements that support the requirements of today's powerful database management systems. The following two sections describe these enhancements.

- The first section describes the enhancements that apply only to NetWare Btrieve 6.1*x*.
- The second section describes enhancements that apply to NetWare Btrieve 6.*x* (that is, to both 6.0*x* and 6.1*x*).

Enhancement to NetWare Btrieve 6.1*x*

This section describes enhancements that apply to NetWare Btrieve 6.1*x*.

New Operations

NetWare Btrieve 6.1*x* supports the following new Btrieve operations.

- Find Percentage and Get By Percentage—Windows-oriented applications can use these new operations for implementing scroll bars.
  - The Find Percentage operation (45) finds either a record's physical location within a file, or the record's position relative to a key path. The location or position is expressed as a percentage value.
  - The Get By Percentage operation (44) retrieves a record by that record's position in the Btrieve file, where the position is based on a percentage value you supply when you call the operation. You can specify whether the position is relative to a specified key path or represents the record's actual physical location in the file.
- Get Direct/Chunk and Update Chunk—NetWare Btrieve 6.1*x* allows applications to operate on portions of a record, called "chunks," rather than on the entire record. Through the new Get Direct/Chunk and Update

Chunk operations, NetWare Btrieve v6.1*x* supports records larger than 64 KB.

Operating on Chunks

NetWare Btrieve 6.1*x* allows you to operate on portions of a record, called *chunks*, rather than on the entire record. NetWare Btrieve 6.1*x* provides the Get Direct/Chunk (23) and Update Chunk (53) functions that work on any file conforming to the Btrieve 6.*x* file format.

Applications can use these chunk operations to build records larger than 64 KB in files that use the NetWare Btrieve 6.0 file format. Applications can use chunk operations to build records longer than 64 KB in any Btrieve 6.*x* file that allows variable-length records.

Note For more information about these operations, refer to the *Btrieve Programmer's Manual* included in the NetWare Btrieve v6.1 Developer's Kit Supplement, which is available through the Novell Professional Developers' Program (PDP).

Supporting Records Larger than 64 KB

When NetWare Btrieve 6.0*x* operates on files containing records larger than 64 KB, the following restrictions apply:

- NetWare Btrieve 6.0*x* does not recognize the Get Direct/Chunk (23) or Update Chunk (53) operations, and therefore can perform only operations that update or retrieve entire records. Those "entire record" operations can retrieve or update only up to 64 KB of a record because the operations' data buffer length parameter, which specifies the amount of data to update or retrieve, can contain a maximum value of 64 KB.

- Because NetWare Btrieve 6.0*x* does not recognize the UpdateChunk (53) operation, you cannot use Btrieve 6.0*x* to modify a record that is larger than 64 KB and still have the resulting record be larger than 64 KB. However, you can use the Update (3) operation to shorten such a record.

- If you use an Update Chunk (53) operation to create a record larger than 64 KB in a file that uses the Btrieve 6.0*x* file format, NetWare Btrieve 6.0*x* cannot retrieve more than the first 64 KB of the record and returns a Status Code 22 (The data buffer parameter is too short) or a Status Code 97 (The communications buffer is too small) to indicate that there is more data to be retrieved.

- If you use an Update Chunk (53) operation in a file that uses the Btrieve 6.0*x* file format, one of the file's Free Space Lists may be left in a state that can cause Btrieve 6.00d or an earlier version of Btrieve to overwrite its own cache. This may result in a server abend.

For more information, see "Accessing Records by Chunks" on page 237 in Appendix A.

Record Locking in Concurrent Transactions

NetWare Btrieve 6.1*x* provides additional concurrency by allowing true record-level locking in concurrent transactions (explained on page 17). An explicit record lock (set by adding a bias to Get and Step operations) does not lock the corresponding data page—it locks only that particular record. Additionally, NetWare Btrieve 6.1x provides the ability to share locks between multiple cursors of the same file in an application.

Successful locking of a record (inside or outside a transaction) guarantees that no other user is able to update or delete that record. In other words, when a user performs an Update (3) or Delete (4) operation on a record that he or she has previously locked, he or she will eventually succeed unless a deadlock situation is detected.

Note A deadlock occurs when two or more users are each waiting on resources that one of the users has not yet released.

If an application opens the same file multiple times, the locks issued by different cursors do not block each other. The cursors are still independent in the sense that they cannot explicitly unlock (using an Unlock [27] operation) the records that were locked by the other cursors.

After a cursor has read a record with a lock, a second cursor can update or delete the record regardless of whether this second cursor locked the record. The first cursor will not get an error until it tries to delete or update that record.

In general, when implementing multiple cursors within a concurrent transaction, locks obtained on a cursor before the application starts a concurrent transaction are automatically released when the cursor becomes an active participant of that transaction.

Note A cursor becomes an active participant of a concurrent transaction if a lock is requested or an Insert/Delete/Update operation is executed with that cursor.

The transaction does not have any effect on the inactive cursors. That is, locks obtained before the transaction are still maintained after ending/aborting the transaction of inactive cursors.

For more information, see “Locks” on page 242 in Appendix A.

New File Structure to Support Accessing Very Long Records

NetWare Btrieve 6.1*x* allows an application to create Btrieve files that contain structures called *Variable-tail Allocation Tables* (*VATs*). VATs give NetWare Btrieve faster access to data residing at large offsets in very long records. VATs also significantly reduce the buffers sizes Btrieve needs to process records in files that use data compression.

For more information, see “Storing Variable-Length Records: Variable Tails and VATs” on page 209 in Appendix A.

Multiple Alternate Collating Sequences

NetWare Btrieve 6.1*x* allows you to specify a separate alternate collating sequence (ACS) for each key in a file. Btrieve files with multiple keys are no longer restricted to having only one ACS.

For more information, see “Alternate Collating Sequence” on page 216 in Appendix A.

Locale-Specific Collating Sequences

NetWare Btrieve 6.1*x* lets you instruct Btrieve to build an ACS that is sensitive to the specified locale's character sorting order. This ability allows Btrieve to sort according to a character set specified by a particular country ID and code page.

For more information, see "Locale-Specific ACS" on page 217 in Appendix A.

Index Balancing

When an index page becomes full, NetWare Btrieve automatically creates a new index page and splits the values in the full page between the two pages. NetWare Btrieve 6.1*x* offers the option of using *index balancing* instead.

When you use index balancing, NetWare Btrieve looks for available space in other index pages associated with the same key each time an index page becomes full. Btrieve then rotates values from the full page onto the pages that have space available. Index balancing increases index page utilization, results in fewer pages, and produces an even distribution of keys among nodes on the same level.

For more information, see "Index Balancing" on page 264 in Appendix A.

Performing Reads While Creating an Index

With NetWare Btrieve 6.1*x*, you can perform reads while a Create Index (13) operation is executing. Previous versions of Btrieve locked the entire file when executing a Create Index (13) operation.

Note

For more information about the Create Index (13) operation, refer to the *Btrieve Programmer's Manual* included in the NetWare Btrieve v6.1 Developer's Kit Supplement, which is available through the Novell Professional Developers' Program (PDP).

New Data Type: sign trailing separate

NetWare Btrieve 6.1*x* allows you to specify a new data type for keys called a `sign trailing separate` (STS). STS is based on a COBOL data type `Numeric sts`.

Basically a numeric data type, it is represented as an ASCII string that is right-justified and padded with zeros. However, the `sign trailing separate` data type stores either a ‘+’ (ASCII 0x2B) or a ‘-’ (ASCII 0x2D) in the right-most byte of its value.

For more information, see “Key Types” on page 224 in Appendix A.

No Currency Change Option

NetWare Btrieve 6.1*x* provides a No Currency Change option on inserts and updates. In previous versions of Btrieve, positioning was reestablished based on a key value of the inserted or updated record.

New Ability to Specify Repeating- or Linked-Duplicatable Keys on Create Operations

On the Create (14) and Create Index (31) operations, NetWare Btrieve 6.1*x* allows you to specify that a key should be created as either a repeating-duplicatable key or a linked-duplicatable key.

For more information, see “Linked-Duplicatable and Repeating-Duplicatable Keys” on page 214 in Appendix A.

Improved NetWare Btrieve DOS and OS/2 Requesters

The NetWare Btrieve DOS and OS/2 Requesters support MAP ROOT drives and NetWare file-level security. Since the MS Windows Requester requires the DOS Requester to access NetWare Btrieve, MS Windows users can also take advantage of these features then they are running in a NetWare 3 or NetWare 4 environment.

When opening files in a NetWare 3.12 environment, Btrieve Requesters provide enhanced performance by reducing the bindery access for each file and reducing network traffic in general.

Note To run NetWare Btrieve 6.1*x* in a NetWare 3.11 environment, you need to load AFTER311.NLM.

The NetWare Btrieve 6.1*x* DOS and OS/2 Requesters have two new features:

- Automatic support for double-byte character environments
- Support for the NetWare Runtime serialized NetWare operating system (the new /C configuration option)

In addition, the DOS Requester also has the following new options:

- Optional compression of data prior to network transmission (the /O configuration option)
- Ability to load another instance of the Btrieve DOS Requester even it one is already loaded (the /L configuration option)

These options are discussed in detail in “DOS Requester Configuration Options” on page 104 in Chapter 4, “Configuring and Using the Requesters.”

Enhanced Communications Programs

An enhanced NetWare Btrieve Message Router (BROUTER.NLM) and a new communications module (BTRVSTUB.NLM) were provided with NetWare Btrieve 6.10b to support NLMs that need to run in the IOEngine of a NetWare SFT III server.

BROUTER.NLM redirects all Btrieve calls from the IOEngine to NetWare Btrieve running in the MSEngine (Mirror Server Engine).

For detailed information about these new programs, see “Communications Programs” on page 29 in Chapter 2, “NetWare Btrieve Architecture.”

For information about running NLMs that require Btrieve in the IOEngine, see “Using NetWare Btrieve with a NetWare SFT III Server” on page 97 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Enhancements to NetWare Btrieve 6.*x*

This section describes enhancements that apply to NetWare Btrieve 6.*x* (that is, to versions 6.0*x* as well as versions 6.1*x*).

Registration with Novell Directory Services

NetWare Btrieve 6.*x* registers with Novell Directory Services (NDS). NDS is a part of NetWare 4 that organizes network resources as objects in a hierarchical tree structure called the Directory. Although the Directory replaces the bindery, NetWare 4 still provides compatibility with previous versions of NetWare through a bindery emulator.

Since NDS organizes the network in terms of objects, NetWare Btrieve registers with NDS as an object of class Btrieve Server. The Btrieve Server class identifies the server name, the type of communications protocol supported, and the object's internet address.

See "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, "Installing and Configuring NetWare Btrieve," for information on installing Btrieve as an object in the Directory.

For more information on NDS, refer to the *Installation and Upgrade* manual (provided with NetWare 4).

New File Format

When creating files, NetWare Btrieve 6.*x* uses a new file format that allows faster data access than was possible with previous Btrieve releases. This format, introduced in Btrieve 6.0 and modified in Btrieve 6.1, is responsible for many of the enhancements and new features available with these releases.

When working with Btrieve files created with different versions of Btrieve, consider the following points:

- New file features supported by Btrieve include the following (discussed in the previous section): Variable-tail Allocation Tables (VATs), multiple alternate collating sequences (ACSs), locale-specific ACSs, and the ability to mark a file as using index balancing.

  NetWare Btrieve 6.1*x* returns a file version of 6.1 if the file specified in a Stat operation (15) or with the BUTIL -STAT command was created with multiple VATs, ACSs, local-specific ACSs, or the index balancing option

Files created with any of these features are created in the Btrieve 6.1*x* format. If NetWare Btrieve 6.1*x* creates a file without any of these features, it creates the file in the Btrieve 6.0*x* file format by default. You can override the 6.0*x* default and create Btrieve 5.*x* files by using the Create Btrieve Files in Pre6.x Format configuration option in the Setup utility, as described on page 70 in Chapter 3, "Installing and Configuring NetWare Btrieve."

- NetWare Btrieve 6.0*x* operates on any file created with NetWare Btrieve v6.1*x* unless that file uses 6.1*x* features that have altered the Btrieve file format (such as VATs and multiple ACSs). This means that NetWare Btrieve 6.0*x* cannot open a file that uses the Btrieve 6.1*x* file format.

Note

NetWare Btrieve 6.0*x* returns a Status Code 30 (The file specified is not a Btrieve file) when you try to open a file created in the Btrieve 6.1x file format.

- Although Btrieve versions earlier than 6.*x* cannot open files that have a 6.*x* format, NetWare Btrieve 6.*x* can open files created with earlier versions of Btrieve.

  For example, NetWare Btrieve 6.0*x* can open files created with Btrieve 5.*x*, and NetWare Btrieve 6.1*x* can open files creates with Btrieve 5.*x* or 6.0*x*. Additionally, NetWare Btrieve 6.1x can write to files using the existing file format—that is, Btrieve 6.1*x* writes to 5.*x* files using the 5.*x* file format and writes to 6.0*x* files using the 6.0*x* file format.

- When NetWare Btrieve6.*x* opens files from earlier versions, it does *not* automatically convert the files to the 6.*x* file format.

  For example, if you use NetWare Btrieve6.1*x* to open a 6.0*x* file, NetWare Btrieve 6.1*x* does not convert the 6.0*x* file to a 6.1*x* format. However, if an application uses Btrieve 6.10a on a 6.0*x* file and creates an index that uses a second ACS or a locale-specific ACS, the file format will be upgraded to a version 6.1*x*, thus preventing the file from being opened by an earlier Btrieve version, which would not recognize these features.

- As indicated above, NetWare Btrieve 5.*x* cannot open files that have a 6.0*x* or 6.1*x* file format.

  If you are using NetWare Btrieve 6.1*x* and you need backward compatibility with Btrieve 5.*x*, use the Btrieve Setup utility option Create Btrieve Files in Pre 6.x Format, described on page 70. This option allows you to create files with NetWare Btrieve 6.1*x* and use then with Btrieve 5.*x*. (Keep in mind, however, that many of the NetWare Btrieve 6.1*x* features require the NetWare Btrieve 6.1*x* file format).

- NetWare Btrieve 6.1*x* allows the creation of an STS data type index in files using the NetWare Btrieve 6.0*x* file format.

  Note If you use NetWare Btrieve 6.00d or an earlier Btrieve version to access a file with an STS data type index, a server abend may occur.

- NetWare Btrieve 6.1*x* supports a longer STS data type than Scalable SQL.

  Make sure that the length of the STS data type does not exceed 15 bytes if you plan to use the NetWare Btrieve 6.1*x* files with Scalable SQL.

If your database contains files created with versions of NetWare Btrieve prior to 6.1*x*, you may want to upgrade them to take advantage of the NetWare Btrieve 6.1*x* features. The Btrieve Rebuild utility converts Btrieve data files to the 6.1*x* file format, as described on page 76 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

New Utilities

NetWare Btrieve 6.*x* includes several utilities to help you use your Btrieve-based applications and monitor your Btrieve operations.

Btrieve Monitor Utility (BTRMON.NLM)

The NetWare Btrieve 6.*x* Monitor utility (BTRMON.NLM) replaces the NetWare Btrieve 5.*x* Console utility (BCONSOLE.NLM).

In terms of functionality, BTRMON is much like BCONSOLE, but it includes new features to help you monitor the performance of your Btrieve-based applications.

Important NLM applications that call NetWare Btrieve must issue a Btrieve Reset before unloading. Failure to do so may lead to a server abend when you try to use the Btrieve Monitor utility to monitor the NLM application’s activity.

For additional information about the Monitor utility, refer to “NetWare Btrieve Monitor Utility” on page 119 in Chapter 5, “Using NetWare Btrieve Utilities.”

Rebuild Utility (BREBUILD.NLM)

The new Rebuild utility allows you to upgrade files created with versions of Btrieve prior to 6.0*x*. Use this utility to upgrade your files to take advantage of the Btrieve 6.1*x* features.

For more information about rebuilding your existing files, refer to page 76 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Note Before running the Rebuild utility (either from the command line or through the Setup utility), you must start NetWare Btrieve v6.1*x*.

Roll Forward Utilities (BROLLFWD.EXE, PBROLL.EXE, and WBROLL.EXE)

The NetWare Btrieve 6.*x* BROLLFWD.EXE file for DOS workstations replaces the NetWare Btrieve 6.0*x* file DBROLL.EXE.

For detailed information about using the Roll Forward utilities, see “NetWare Btrieve Roll Forward Utility” on page 171 in Chapter 5, “Using NetWare Btrieve Utilities.”

Online Backups

Through a feature called *continuous operation*, Btrieve 6.*x* now lets you back up Btrieve files while they are open and in use. This feature is important for applications that conduct transactions 24 hours a day.

When you enable the continuous operation feature, Btrieve opens the original file in read-only, sharable mode to allow backup utilities to access the file’s static image. Any changes to the original file that occur during the backup are stored in a temporary file called a *delta* file. When the backup ends, Btrieve automatically updates the original file with the changes from the delta file and deletes the temporary delta file.

Concurrent Transactions

Concurrent transactions allow one or more applications (or instances of the same application) to run multiple transactions simultaneously for the same Btrieve file, if that file uses the Btrieve 6.*x* file format.

Versions of Btrieve earlier than 6.0*x* support only one type of transaction: *exclusive*. When an application reads, updates, inserts, or deletes a record from a file in an exclusive transaction, Btrieve locks the entire file for the duration of the transaction. This type of locking is known as *file-level transaction locking*.

Once a file is locked in an exclusive transaction, other users can still read the file, provided they are not involved in exclusive transactions and are not attempting to lock the file themselves. They cannot, however, make any changes to the file, and they cannot see changes made to the file during a transaction until that transaction ends (regardless of which version of Btrieve was used to create the file originally).

Note

This is true only for local clients—that is, clients on the same workstation running under the same Btrieve engine. However, if the exclusive transaction affects pre-v6.0 files, remote clients—clients on different workstations, running under different Btrieve engines—*can* see the changes to those files before the transaction ends.

In addition to supporting exclusive transactions, Btrieve 6.*x* supports a new type of transaction: *concurrent*. When a Btrieve 6.*x* file is included in a concurrent transaction, modifications cause Btrieve to lock only the page (or pages, if the record is variable length) that contains the record, as well as its associated index pages.

This allows other users to modify or include the same file in their own concurrent transaction, as long as no concurrent transaction has already locked the pages that contain the records to be modified (or any affected index pages).

Concurrent transactions have the following additional features:

- Locked pages remain locked for the duration of the transaction.

- Other clients can read the data on the locked pages; however, they cannot lock the pages (by updating).

- If the transaction simply reads a record, Btrieve does *not* lock the corresponding page (unless a read lock is applied). Other users can apply an explicit read lock to an individual record on the same page if that record is not locked by the first user. (Previous version of Btrieve converted explicit read locks to a page-level lock.)

- Multiple cursors for the same file share locks. Other cursors can lock the same record locked by the first cursor.

- Other clients cannot see the changes to a file in a transaction until the transaction ends. (That way, if a system failure occurs before the transaction completes, other clients will not have read false data—that is, data that will be rolled back.)

Note

For compatibility, Btrieve 6.*x* continues to support exclusive transactions, but with one difference from previous versions: if the exclusive transaction affects

Btrieve 6.*x* files, other clients cannot see the changes to those files until the transaction ends.

For more information, see “Using Btrieve Transactions” on page 238 in Appendix A.

Shadow Paging

Versions of Btrieve prior to 6.0*x* used *pre-imaging* to protect files from corruption in case of a system failure. Btrieve 6.x also uses this method to protect files that employ the pre-6.0*x* file format.

Note

Before updating a file using the pre-imaging feature, Btrieve creates a temporary pre-image file. This file contains the pages to be updated from the original file. Btrieve then performs the update on the original file. If the system fails during the update, Btrieve can restore the original file using the pre-image file. For more informations about pre-image files, see page 246 in Appendix A.

For files using the Btrieve 6.*x* file format, a new Btrieve 6.*x* page handling technique called *shadow paging* replaces pre-imaging.

When a user needs to change a page (either inside or outside a transaction), Btrieve creates a shadow page—a virtual copy of the original page. Btrieve then selects and locks a free physical location in the Btrieve file for the shadow page, but it does not write the shadow page to the new location at this point.

Instead, it makes the requested changes to the shadow page (rather than to the original page) while the shadow page is still in memory. After making the requested changes, Btrieve writes the shadow page to the new physical location.

When the changes are committed (either when the operation is complete or the transaction ends), Btrieve designates the shadow page as the current page, and the original page becomes available for reuse. If a system failure occurs before the changes are committed, Btrieve drops the shadow page, and the current page remains in its original condition.

When a user first attempts to change a page inside a transaction, Btrieve creates a shadow page that corresponds to the original logical page. While inside the transaction, all changes the client makes are actually made to the shadow page.

If other users need to access the same logical page, they will see the original (unchanged) page—that is, they do not see the first client's uncommitted changes. Shadow paging thus enhances reliability because the original file is always valid and internally consistent.

For more information about shadow paging, see page 245 in Appendix A.

New Caching Algorithms

Btrieve 6.*x* provides new caching algorithms that improve memory management for concurrent users. These new algorithms include hashing search methods for improved access, concurrent sharing of a single cache, and use of existing cache across operations.

Better Use of Large Data Files

Btrieve 6.*x* provides faster access and more efficient use of large data files than previous versions of Btrieve provided. With Btrieve 6.*x*, you can create additional indexes for large data files more quickly than in previous versions, and new merge sorts take advantage of whatever cache is available.

More Key Segments Allowed

Btrieve 6.*x* has increased the maximum number of key segments allowed in a file, supporting up to 119 key segments in files with a page size of 4,096 bytes. (Versions of Btrieve earlier than 6.0*x* supported a maximum of 24 key segments.)

The maximum number of key segments you can define for a file depends on the file's page size, as indicated in the following table:

Page Size	Maximum Key Segments
512	8
1,024	23
1,536	24
2,048	54
2,560	54
3,072	54
3,584	54
4,096	119

For more information, see "Segmented and Nonsegmented Keys" on page 212 in Appendix A.

Adding and Dropping Any Index

Btrieve 6.*x* supports adding and dropping any index. In versions prior to Btrieve 6.0*x*, you could add and drop only supplemental indexes—that is, only those indexes that were not originally produced during a Create operation (14). Also, you can drop indexes without renumbering the remaining indexes.

This feature requires that an application take extra precautions when accessing a Btrieve file that may be shared with other applications. To be certain that the file being opened contains all the keys application 1 is expecting and that those keys are in the proper order, application 1 should preform a Stat operation(15) before using a particular index.

For more information, see "Indexes" on page 231 in Appendix A.

Specific Key Numbers Allowed When Creating a File or Index

Btrieve 6.*x* allows an application to assign specific key numbers when creating a 6.*x* format file with indexes, or when creating an index for a preexisting 6.*x* file.

Btrieve 6.*x* files are not required to number keys consecutively; they may have gaps between key numbers. When a key is added, Btrieve by default assigns the lowest available key number to that index.

However, some applications may require a different key number than the one assigned by default. For this reason, Btrieve 6.*x* provides a way to specify which key number the Create Index operation (31) assigns to the index being created.

This capability complements the ability to delete a key and not have Btrieve renumber all keys that have a key number greater than that of the deleted key (described in the next section).

For example, assume that an application drops an index and instructs Btrieve not to renumber higher-numbered keys. If a user then clones the affected Btrieve file without assigning specific key numbers, the cloned file would have different key numbers than the original.

For more information, see “Keys” on page 211 in Appendix A.

Optional Renumbering of Keys

With Btrieve 6.*x*, the Drop Index operation (32) allows an application to specify whether to renumber the remaining keys when dropping a key from a Btrieve file. (The file must use the Btrieve 6.*x* file format.)

When an application dropped a key in earlier versions of Btrieve, Btrieve renumbered the remaining keys that had a key number higher than that of the dropped key. To renumber these keys, Btrieve decremented each higher key number by one so that no key numbers were skipped.

In 6.*x*, an application can specify that when an index is dropped, no automatic renumbering should take place. In this case, the remaining higher numbered keys retain their original key numbers after Btrieve drops the specified key.

For more information, see “Indexes” on page 231 in Appendix A.

Enhanced Support for Case-Insensitive Keys

Btrieve 6.*x* allows you to create case-insensitive keys without using an alternate collating sequence (ACS). When creating a file, you can use the key specifications to designate case-insensitive `string`, `lstring`, or `zstring` key types.

When using case-insensitive keys, Btrieve treats lowercase and uppercase letters the same. For example, the lowercase letter *a* and the uppercase letter *A* are treated as the same letter for sorting purposes.

For more information, see “Case-Insensitive and Case-Sensitive Keys” on page 216 in Appendix A.

Enhancement to autoincrement Key

In Btrieve 6.*x*, you can initialize the value of a field in every record of a file to zero and later add an index of type `autoincrement`. This feature allows you to prepare for an `autoincrement` key without actually building the index until you need it.

When you add the index, Btrieve changes the zero values in each field appropriately, beginning its numbering with a value equal to the greatest value currently defined in the field, plus one. If nonzero values exist in the field, Btrieve does not alter them. However, Btrieve returns an error if nonzero duplicate values exist in the field.

In Btrieve versions earlier than 6.0*x*, Btrieve returned a duplicates error if a user added an `autoincrement` key as an index and the field being indexed contained more than one zero.

Note

If an error occurs on a Create Index operation (31) after any values have been altered, those values remain altered.

For more information, see “Key Types” on page 224 in Appendix A.

Reserved Space for Duplicate Pointers

Btrieve 6.*x* allows reserving duplicate pointers on data records for linked duplicate keys. When creating a Btrieve 6.*x* file, an application can reserve space in the data records for extra, unused duplicate pointers.

Later, when an application adds an index that allows duplicate values, Btrieve stores pointers to those duplicate values in the reserved space (unless the Repeating Duplicates key attribute has been specified).

Key-Only File Modification

With Btrieve 6.*x*, you can update and delete records in key-only files. In versions of Btrieve prior to 6.0*x*, you could only insert records in this type of file.

For more information, see “Key-Only Files” on page 204 in Appendix A.

New Stat Option

A new option in the Stat operation (15) allows an application to obtain additional information such as a file’s Btrieve version and the number of unused duplicate pointers. This feature also works with files created with earlier versions of Btrieve.

Locking in Extended Operations

Unlike previous versions of Btrieve, 6.*x* allows an application to use locks on the extended operations: Get Next Extended (36), Get Previous Extended (37), Step Next Extended (38), and Step Previous Extended (39).

For more information, see “Locks” on page 242 in Appendix A.

Support for Referential Integrity

Btrieve 6.*x* supports the use of referential integrity (RI) constraints created through the Novell® Scalable SQL™ or client-based SQL relational data access system. RI ensures that dependent data stays synchronized throughout the database.

No Btrieve operations currently exist to manipulate RI constraints directly. You must use either Scalable SQL or the client-based version of SQL.

chapter 2 NetWare Btrieve Architecture

If you are new to the Btrieve® record management system, you should read this chapter before installing and configuring NetWare Btrieve. It provides a general introduction to the NetWare Btrieve components and how they work with Btrieve-based applications.

If you have used Btrieve before, you may want to skip to the example diagrams of Btrieve architecture, which begin in “Examples of Btrieve Architecture” on page 38.

This chapter includes the following sections:

- “Components of NetWare Btrieve” on page 28
- “NetWare Btrieve Applications on a Server” on page 33
- “Btrieve Applications on a Workstation” on page 35
- “Examples of Btrieve Architecture” on page 38

Components of NetWare Btrieve

The following sections describe the major components of server-based Btrieve:

- "Server-Based Record Manager" on page 28
- "Communications Programs" on page 29
- "Workstation Requesters" on page 31
- "NetWare Btrieve Utilities" on page 32
- "Novell Directory Services Support Utility" on page 33

For a list of the files that comprise NetWare Btrieve, see "NetWare Btrieve 6.1x Program Files" on page 53 in Chapter 3, "Installing and Configuring NetWare Btrieve."

Server-Based Record Manager

The NetWare Btrieve server-based Record Manager (BTRIEVE.NLM) must be loaded on every server that accesses Btrieve files. The Btrieve NLM, which consists of a library of Btrieve function calls, handles these tasks:

- Performs disk I/O for Btrieve files at the server where it resides
- Provides concurrency and integrity controls on the server where it resides
- Logs all Btrieve requests that result in changes to a file (if logging is enabled for that file)

Communications Programs

Btrieve provides these communications programs, described in more detail in the following sections:

- BROUTER.NLM and BDROUTER.NLM—BROUTER, the NetWare Btrieve Message Router, allows server-based Btrieve applications to access Btrieve databases on remote servers. BDROUTER provides Directory Services support in a NetWare 4 environment.

- BSPXCOM.NLM—The SPX communications agent.

- BSPXSTUB.NLM—Module that allows you to use the Btrieve Monitor utility (BTRMON.NLM) when BSPXCOM is not loaded.

- RSPXSTUB.NLM—Module that resolves external references for the Btrieve Monitor utility when BSPXCOM is not loaded.

  If you want to use the Btrieve Monitor utility to monitor outgoing requests generated by BROUTER to another server, and you do not want to load BSPXCOM, load RSPXSTUB instead of BSPXSTUB.

Btrieve Message Routers

The Btrieve Message Routers (BROUTER.NLM or BDROUTER.NLM) handles outgoing requests from your server to a remote server.

The Message Routers allow a Btrieve application running as an NLM™ on the server to communicate with remote servers on which other Btrieve NLMs are loaded. Also, the Message Routers maintain transaction concurrency controls during transactions involving Btrieve files on more than one server.

- BROUTER.NLM—This is the default Message Router, which is loaded through the NetWare® INSTALL utility. (For more information about the NetWare INSTALL utility, refer to “Using the NetWare INSTALL Utility” on page 58 in Chapter 3, “Installing and Configuring NetWare Btrieve.”)

- BDROUTER.NLM—This is a Btrieve 6.*x* Message Router that provides Directory Services support.

Note

If you want Directory Services support, you need to unload BROUTER.NLM and load BDROUTER.NLM. Although you cannot use BDROUTER.NLM in a NetWare 3 environment, BDROUTER can communicate with both NetWare 3 and NetWare 4 servers.

If a server-based Btrieve application needs to access files on a remote server, the Message Router must be loaded on the local server.

When you request files from the remote server, the Message Router sends that request to BSPXCOM (discussed in the next section) on the remote server. BSPXCOM routes your request to the Btrieve NLM on the remote server and then sends the response back to the Message Router on the local server.

BSPXCOM

BSPXCOM handles incoming requests to the Btrieve NLM from a remote source. The remote source could be a requester at a workstation or the Message Router on another server. BSPXCOM must be loaded on all servers that support remote requests.

If no workstations or remote servers make requests to the local Btrieve NLM, you may not want to have BSPXCOM loaded at the local server.

For example, assume the local Btrieve NLM receives calls only from other NLMs running on the same server. In this case, you could choose not to load BSPXCOM for security reasons. (Not loading BSPXCOM allows you to restrict applications other than the ones on the local server from accessing your Btrieve files.)

BSPXSTUB and RSPXSTUB

If you do not load BSPXCOM and want to use the Btrieve Monitor utility (BTRMON.NLM), you must load either the BSPXSTUB or the RSPXSTUB communications module. These modules resolve external references for the Monitor utility that BSPXCOM would otherwise resolve.

Use the following guidelines to determine whether you need BSPXSTUB or RSPXSTUB:

- If you want to use the Btrieve Monitor utility but do not want to load BSPXCOM, and the NLMs on the local server are accessing Btrieve files only on that server, load BSPXSTUB at the local server.

- If you want to use the Btrieve Monitor utility to monitor outgoing requests generated by the Message Router to a remote server and you do not want to load BSPXCOM, load RSPXSTUB instead of BSPXSTUB at the local server.

Note

The Btrieve Monitor utility's Communication Statistics option (discussed on page 135 in Chapter 5, “Using NetWare Btrieve Utilities”) displays SPX communication statistics. The communications module you load affects the statistics displayed.

For example, if you load BSPXCOM, you see incoming and outgoing SPX statistics for BSPXCOM. If you load BSPXSTUB, you see all zeros. If you load RSPXSTUB, you see incoming and outgoing SPX communication statistics from the Message Router.

Workstation Requesters

NetWare Btrieve provides the following requesters for applications running on the workstation:

- BREQUEST.EXE—DOS Requester
- BTRCALLS.DLL—OS/2 Requester
- WBTRCALL.DLL—MS Windows Requester

A NetWare Btrieve Requester must be loaded at each workstation that makes Btrieve requests. The requester receives Btrieve requests from an application and relays them via BSPXCOM to the Btrieve NLM running on the server.

After the Btrieve NLM processes the request, BSPXCOM sends the results back to the requester, which forwards them to the application.

For information on starting the requester in each operating environment, refer to Chapter 4, “Configuring and Using the Requesters.”

NetWare Btrieve Utilities

NetWare Btrieve provides the following utilities for Btrieve file management:

- BSETUP.NLM—The Btrieve Setup utility, which you use to change the settings of the NetWare Btrieve configuration options. (See “Configuring NetWare Btrieve” on page 63 in Chapter 3, “Installing and Configuring NetWare Btrieve.”)

- BREBUILD.NLM—The Btrieve Rebuild utility, which you use to convert existing Btrieve 5.*x* files to the Btrieve 6.*x* file format. (As explained in “Rebuilding Existing NetWare Btrieve Files” on page 76 in Chapter 3, “Installing and Configuring NetWare Btrieve,” you can choose to run the Rebuild utility from within the Setup utility.)

- BTRMON.NLM—The Btrieve Monitor utility, which you use to monitor the activity of NetWare Btrieve at the server.

- BUTIL.NLM—The Btrieve Maintenance utility, which you use to import and export Btrieve data and to transfer data from one Btrieve file to another. This utility also lets you enable and disable continuous operation on your Btrieve files.

- BROLLFWD.EXE for DOS, PBROLL.EXE for OS/2, and WBROLL.EXE for MS Windows—The Btrieve Roll Forward utilities recover changes made to a Btrieve file between the time of the last backup and a system failure. The Btrieve Roll Forward utilities are for workstation use only.

For more information about the Btrieve Setup and Rebuild utilities, refer to Chapter 3, “Installing and Configuring NetWare Btrieve.” For information about the Btrieve Monitor, Maintenance, and Roll Forward utilities, refer to Chapter 5, “Using NetWare Btrieve Utilities.”

Novell Directory Services Support Utility

NetWare Btrieve 6.1*x* includes a Novell Directory Services™ (NDS™) Support utility that allows you to register Btrieve with NDS as an object of the class Btrieve Server. Using this utility, you can install or remove a Btrieve Server object from the Directory.

You can run the NDS Support utility (BDIRECT.NLM) interactively, through the Btrieve Setup utility, or from the command line.

For more information about the NDS Support utility, refer to “Registering NetWare Btrieve with the Novell Directory Services” on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

NetWare Btrieve Applications on a Server

A NetWare Btrieve application running on a server (that is, a Btrieve-based application running as an NLM) can access data on the local server or on a remote server, as follows:

- Server Application Accessing Local Data—The application makes a request for Btrieve data stored on the local server. The Btrieve NLM on the server where the Btrieve data request originated processes the request.

- Server Application Accessing Remote Data—The application makes a request for Btrieve data stored on a remote server. The Btrieve NLM on the remote server processes the request.

The following sections describe the events that occur when a Btrieve-based application running as an NLM on a NetWare server makes local and remote requests.

Server Application Accessing Local Data

When an application running on a server accesses data stored on that server, the Btrieve NLM must also be loaded on that server. The following steps describe accessing data stored on the local server:

1. The application sends a request to the Btrieve NLM. If the Message Router (BROUTER.NLM) is not loaded, the call is made directly to the exported entry point of Btrieve on the local server.

   If the Message Router is loaded, it relays the call to Btrieve on the local server.

2. Btrieve processes the request using the Btrieve library of function calls.

3. If the Message Router is not loaded, Btrieve returns the appropriate data and status code directly to the calling application.

   If the Message Router is loaded, Btrieve returns the data and status code to the Message Router. The Message Router then transports the data and status code to the calling application.

Server Application Accessing Remote Data

When an application running on a local server accesses data stored on a remote server, both the Message Router and the Btrieve NLM must be loaded on the local server, and BSPXCOM and the Btrieve NLM must be loaded on the remote server. The following steps describe accessing data on a remote server:

1. The application (running on the local server) makes a request to access a Btrieve file stored on a remote server.

2. The Message Router on the local server detects that the request is for a remote Btrieve file and sends the request to BSPXCOM on the remote server.

3. BSPXCOM (remote) relays the request to the Btrieve NLM (remote) by making Btrieve function calls.

4. The Btrieve NLM (remote) returns the appropriate data and status code to BSPXCOM (remote).

5. BSPXCOM (remote) returns the data and status code to the local server, where the Btrieve request originated.

6. The Message Router (local) returns the results to the calling application (local).

   The Message Router places the results in the application's memory at the location designated by the parameters passed to Btrieve in the function call. Control then returns to the calling application.

Btrieve Applications on a Workstation

A Btrieve application running on a workstation can access local, remote, or both local and remote data as follows:

- Workstation Application Accessing Local Data—The application makes a request for Btrieve data stored on the workstation. Client-based Btrieve processes the request on the same workstation where the request originated.

- Workstation Application Accessing Remote Data—The application makes a request for Btrieve data stored on a remote server. The request is sent to the requester, which passes the request to the Btrieve NLM on the remote server. The Btrieve NLM processes the request.

- Workstation Application Accessing Local and Remote Data—The application makes a request for Btrieve data stored on either the local workstation or on a remote server. The request for data is intercepted by the Btrieve Requester on the workstation.

  The requester determines whether the data is stored on the workstation or on a remote server and routes the appropriate request to client-based Btrieve on the workstation or to the Btrieve NLM on a remote server.

The following sections explain what happens when your workstation application makes local and remote requests.

Workstation Application Accessing Local Data

When an application accesses data stored locally on the workstation, client-based Btrieve must be loaded on the workstation.

The following steps describe accessing data on the workstation:

1. The application makes a Btrieve request using a function call.

2. The interface code that you link with your application makes the call to client-based Btrieve. (Novell® provides the interface code.)

   Note: In a MS Windows or OS/2 environment, you must import the function definition.

3. Client-based Btrieve processes the request using the Btrieve library of function calls.

4. Client-based Btrieve returns the appropriate data and status code directly to the calling application.

Workstation Application Accessing Remote Data

When an application accesses data stored on a remote server from a workstation, the Btrieve Requester must be loaded on the workstation, and BSPXCOM and the Btrieve NLM must be loaded on the server.

The following steps describe accessing data stored on a server from an application running on a workstation:

1. The application makes a Btrieve request using a function call.

2. The interface code linked with the application makes the call to the requester. (Novell provides the interface code.)

   Note: In a MS Windows or OS/2 environment, you must import the function definition.

3. The requester packages the request into a network message and routes the message to BSPXCOM on the remote server.

4. BSPXCOM receives the network message, validates the parameters, and then executes the request by making function calls to the Btrieve NLM.

5. The Btrieve NLM processes the request and returns the results to BSPXCOM.

6. BSPXCOM forwards the results to the requester at the workstation.

7. The requester returns the appropriate data and status code to the parameter variables in your application’s memory and returns control to your application.

Workstation Application Accessing Local and Remote Data

When an application accesses local and remote data from a workstation, the Btrieve Requester and client-based Btrieve must be loaded on the workstation, and BSPXCOM and NetWare Btrieve must be loaded on the server.

The following steps describe accessing local and remote data from an application running on the workstation:

1. The application makes a Btrieve request using a function call.

2. The interface code linked with the application makes the call to the requester. (Novell provides the interface code.)

   Note: In a MS Windows or OS/2 environment, you must import the function definition.

3. The requester determines whether the server or the workstation should receive the request.

4. This step varies, depending on whether the requested data is on the workstation or on a remote server:

   - *If the requested data is on the workstation,* the requester sends the request directly to client-based Btrieve. Btrieve processes the request using the Btrieve library of function calls. Client-based Btrieve returns the appropriate data and status code directly to the calling application.

   - *If the requested data is on a remote server,* the requester packages the request into a network message and routes the message to BSPXCOM on that server. BSPXCOM receives the network message, validates the parameters, and then executes the request by making function calls to the Btrieve NLM.

The Btrieve NLM processes the request and returns the results to BSPXCOM. BSPXCOM forwards the results to the requester at the workstation. The requester returns the data and status code to the calling application.

Examples of Btrieve Architecture

The diagrams in this section demonstrate how different Btrieve applications require different Btrieve components. This section discusses the following examples:

- “Server Application Using the NetWare Btrieve NLM” on page 39
- “Server Application Using the NetWare Btrieve Message Router” on page 40
- “Server Application Using the NetWare Btrieve Message Router and BSPXCOM” on page 41
- “Workstation Application Using the Requester and Client-Based Btrieve” on page 42
- “Server Application Using RSPXSTUB” on page 43
- “Server Application Using BSPXSTUB” on page 44
- “Server Application Using Scalable SQL” on page 45

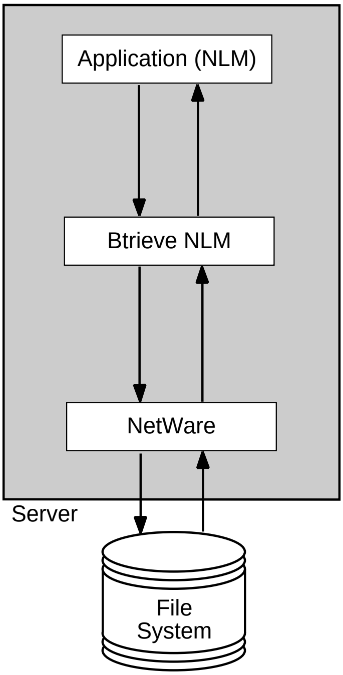
Note

The following examples indicate remote requests with dashed lines and local requests with solid lines.

Server Application Using the NetWare Btrieve NLM

Figure 1 shows an application accessing Btrieve data stored on the local server. The application is making local requests to the local Btrieve NLM. Note that BSPXCOM is not loaded because there are no incoming requests to the Btrieve NLM from another server or workstation.

Figure 1
Server Application Using the NetWare Btrieve NLM

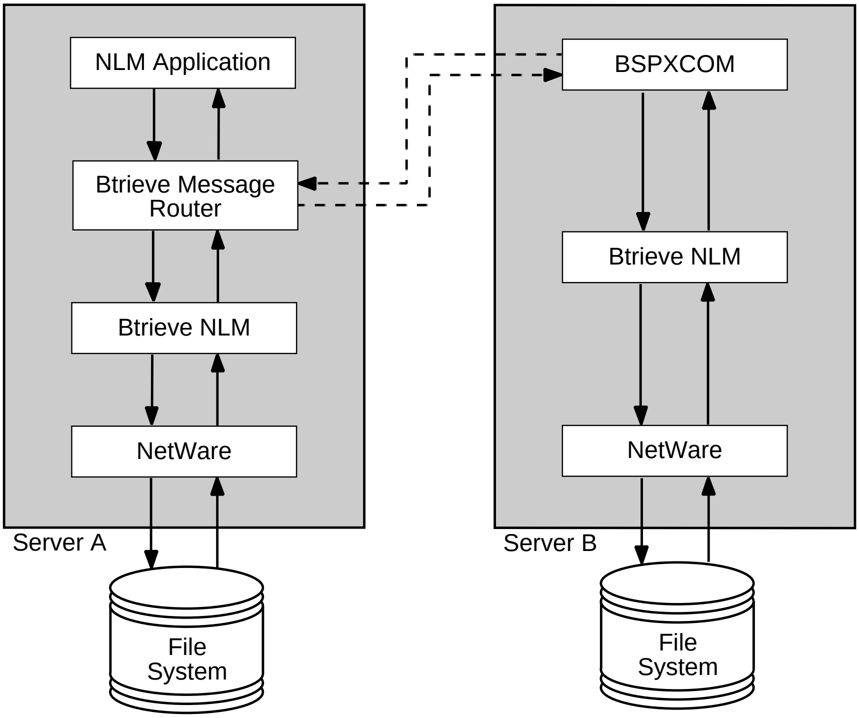


Server Application Using the NetWare Btrieve Message Router

Figure 2 shows an application running on Server A. It is making requests to the local Btrieve NLM (Server A) and to a remote Btrieve NLM (Server B) via the Btrieve Message Router.

The Message Router handles outgoing requests from Server A to the remote Server B. The Message Router must be loaded on Server A in order to send the requests to Server B. BSPXCOM must be loaded on Server B to accept incoming requests from the Message Router.

Figure 2
Server Application Using the NetWare Btrieve Message Router

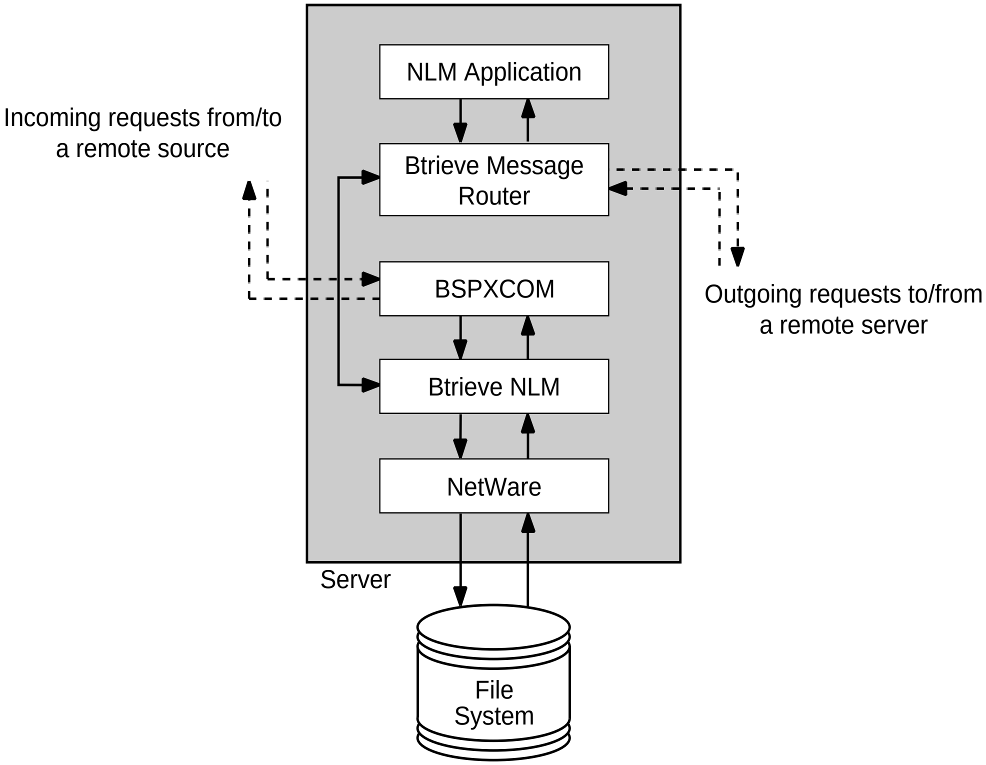


Server Application Using the NetWare Btrieve Message Router and BSPXCOM

Figure 3 illustrates a server application that requires both the Btrieve Message Router and BSPXCOM.

In this figure, the Message Router handles outgoing requests from the local server to a remote server. BSPXCOM handles incoming requests to the Btrieve NLM from a remote source (either a requester at a workstation or the Message Router on another server).

Figure 3
Server Application Using the NetWare Btrieve Message Router and BSPXCOM


NLM Application
Incoming requests from/to a remote source
Btrieve Message Router
BSPXCOM
Outgoing requests to/from a remote server
Btrieve NLM
NetWare
Server
File System

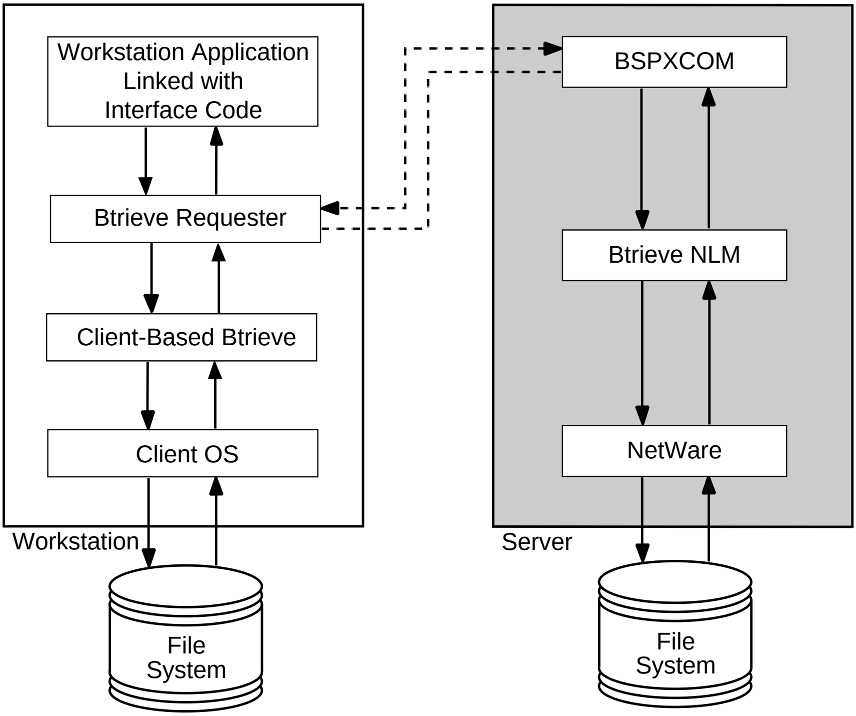
The server is supporting incoming requests from a remote source and outgoing requests to a remote server.

Workstation Application Using the Requester and Client-Based Btrieve

Figure 4 shows an application running on a workstation. The application is accessing local data via client-based Btrieve and remote data via the requester. The requester passes requests for local data to client-based Btrieve.

In this environment, the requester on the workstation performs the same function as the Message Router in Figure 3. BSPXCOM handles incoming requests to the Btrieve NLM from a remote source.

Figure 4
Workstation Application Using the Requester and Client-Based Btrieve

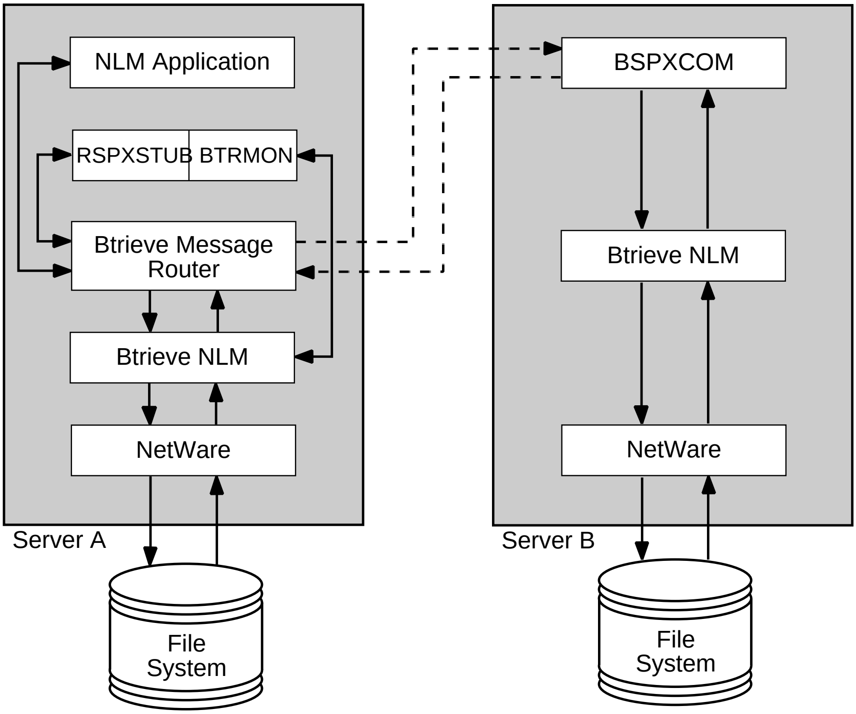


Server Application Using RSPXSTUB

Figure 5 shows an application running on Server A. It is accessing both local and remote data. The Btrieve Monitor utility (BTRMON.NLM) is running on Server A. To run the Monitor utility, Server A must have BSPXCOM, BSPXSTUB, or RSPXSTUB loaded.

Since the Btrieve NLM on Server A is not accepting any incoming requests from workstations or remote servers, BSPXCOM is not loaded. Instead, RSPXSTUB is loaded on Server A. This allows users to see the Btrieve Message Router's communication statistics through the Monitor utility's Communication Statistics option.

Figure 5
Server Application Using RSPXSTUB



Server Application Using BSPXSTUB

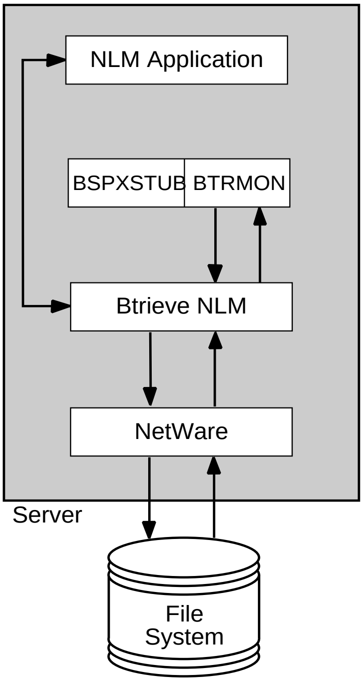
Figure 6 shows an application running on the server. The application is accessing local data. The Btrieve Monitor utility (BTRMON.NLM) is also running on the server.

Since the Btrieve NLM is not accepting any incoming requests, BSPXCOM is not loaded. However, to run the Monitor utility, the server must have BSPXCOM, BSPXSTUB, or RSPXSTUB loaded. Since the Btrieve Message Router is not loaded, BSPXSTUB is the appropriate communications module to have loaded.

Note

When BSPXSTUB is loaded, the Communication Statistics option in the Btrieve Monitor utility shows zeros for the SPX statistics, since there is no SPX activity.

Figure 6
Server Application Using BSPXSTUB

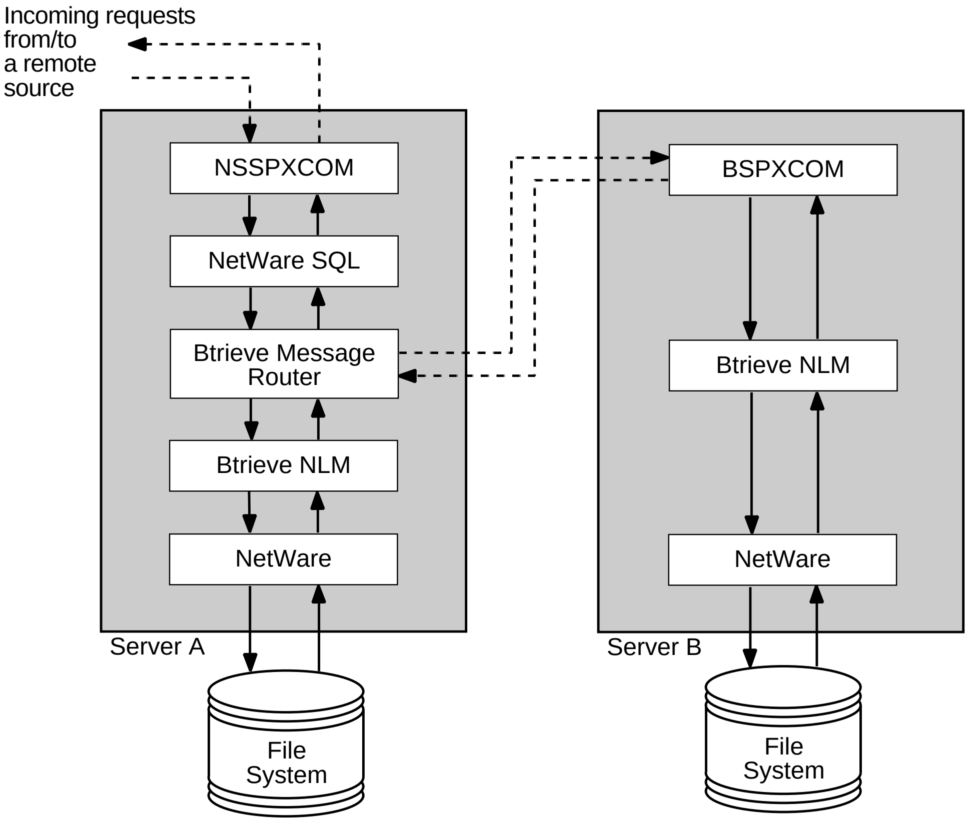


Server Application Using Scalable SQL

Figure 7 shows Scalable SQL running on Server A. Since Scalable SQL needs to access data on both Server A and Server B, the Btrieve Message Router is loaded on Server A.

In addition, NSSPXCOM is loaded on Server A because Scalable SQL is also responding to requests from a Scalable SQL Requester running on a remote workstation. (NSSPXCOM provides the same functionality for Scalable SQL as BSPXCOM provides for Btrieve.)

Figure 7
Server Application Using Scalable SQL



chapter 3

Installing and Configuring NetWare Btrieve

This chapter describes how to install and configure the Btrieve® record management system on a NetWare network. It includes the following sections:

- “Overview of Installation and Configuration” on page 48
- “Preparing for Installation” on page 49
- “Installing NetWare Btrieve” on page 52
- “Detecting and Repairing PAT and File Corruption” on page 59
- “Configuring NetWare Btrieve” on page 63
- “Starting and Stopping NetWare Btrieve” on page 75
- “Rebuilding Existing NetWare Btrieve Files” on page 76
- “Registering NetWare Btrieve with the Novell Directory Services” on page 93
- “Using NetWare Btrieve with a NetWare SFT III Server” on page 97
- “Using NetWare Btrieve with NetWare Runtime” on page 98

Overview of Installation and Configuration

In general, follow these steps to install and configure the NetWare Btrieve™ 6.1*x* components:

Procedure

1. **Prepare for the installation by removing any existing pre-image files and unloading any existing versions of Btrieve.**

   For more information, refer to “Preparing for Installation” on page 49.

2. **Ensure that your system meets the system requirements for running NetWare Btrieve 6.1*x*.**

   For more information, refer to “System Requirements for Installation” on page 52.

3. **Install NetWare Btrieve 6.1*x*.**

   For more information, refer to “Using the NetWare INSTALL Utility” on page 58.

4. **Repair any corruption that may have occurred to the Page Allocation Tables (PAT) pages in your Btrieve files or to the Btrieve files themselves.**

   For more information, refer to “Detecting and Repairing PAT and File Corruption” on page 59.

5. ***Optional*: Use the Setup utility to perform any of the following operations:**

   - Customize the NetWare Btrieve 6.1*x* configuration options, as described in “Configuring NetWare Btrieve” on page 63.

   - Convert existing Btrieve 5.*x* files to the 6.*x* file format, as described in “Rebuilding Existing NetWare Btrieve Files” on page 76.

   - Register NetWare Btrieve with the NDS, as described in “Registering NetWare Btrieve with the Novell Directory Services” on page 93.

6. **Start NetWare Btrieve 6.1*x*.**

   For more information, refer to “Starting and Stopping NetWare Btrieve” on page 75.

7. ***Optional*: Configure NetWare Btrieve for use with an SFT server.**

   For more information, refer to “Using NetWare Btrieve with a NetWare SFT III Server” on page 97.

8. ***Optional*: Configure NetWare Btrieve for use with the NetWare Runtime operating system.**

   For more information, refer to “Using NetWare Btrieve with NetWare Runtime” on page 98.

Preparing for Installation

If you have an earlier version of NetWare Btrieve (such as 5.*x* or 6.0) installed on your system, perform the following operations *before* you run the NetWare INSTALL program:

◆ Remove any existing pre-image files (files with a .PRE extension) in your database.

◆ Unload any existing, earlier versions of Btrieve currently on your system.

Checking for and Removing Extraneous Pre-Image Files

Versions of Btrieve prior to 6.0*x* used pre-imaging to protect files from corruption in case of a system failure. Btrieve 6.x also uses this method to protect files that employ the pre-6.0*x* file format.

Note For files using the Btrieve 6.*x* file format, a new Btrieve 6.*x* page handling technique called shadow paging replaces pre-imaging, as described in “Shadow Paging” on page 19 in Chapter 1, “Introduction to NetWare Btrieve.”

Before updating a file using the pre-imaging feature, Btrieve creates a temporary pre-image file. This file, with a .PRE extensions, contains the pages to be updated from the original file. Btrieve then performs the update on the original file. If the system fails during the update, Btrieve can restore the original file using the pre-image file.

If you have any extraneous pre-image (.PRE) files in your database, you must remove them before installing and starting Btrieve 6.1*x*.

Important

If you do not remove the extraneous .PRE files remaining from using earlier version of Btrieve before installing Btrieve 6.1*x*, file corruption may occur.

The following procedure explains how to check for and remove extraneous .PRE files. Note that you *must* have Btrieve 5.*x* or 6.0 loaded before performing these steps.

Procedure

1. **Using your existing version of Btrieve (such as 5.*x* or 6.0), open the Btrieve data file corresponding to the .PRE file that you want to remove in Exclusive mode.**

2. **Perform a Get First (12) operation.**

3. **Perform an Update (3) operation (this will *not* change the first record).**

4. **Close the Btrieve data file.**

   This should delete the associated .PRE file.

5. **Any .PRE files that remain for the designated file after you perform Steps 1 through 4 are extraneous. Delete them now.**

6. **Repeat Steps 1 through 5 to find and delete any other extraneous .PRE files.**

Once you have removed the extraneous .PRE files from your database, remove the earlier version of Btrieve from your system, as described in the following section.

Note

If you need to preserve your Btrieve 5.*x* files, you can use Btrieve 6.*x* to read them.

Unloading an Earlier Version of NetWare Btrieve

To unload an earlier version of NetWare Btrieve from a server's memory before installing version 6.1*x*, enter the following command at the server console, or at a workstation running RCONSOLE:

```
bstop
```

The BSTOP command runs a NetWare command file (BSTOP.NCF) that unloads the following modules in the order shown:

1. BROUTER.NLM
2. BSPXCOM.NLM
3. BTRIEVE.NLM

Note BSTOP.NCF does not unload BSPXSTUB, RSPXSTUB, or BDROUTER. To unload these modules, enter the UNLOAD command followed by the module name, as in the following example:

```
unload bspxstub | rspxstub | bdrouter
```

Installing NetWare Btrieve

This section discusses the following topics:

- "System Requirements for Installation"
- "NetWare Btrieve 6.1x Program Files" on page 53
- "Using the NetWare INSTALL Utility" on page 58

System Requirements for Installation

Make sure your system has the following:

Checklist

☐ NetWare® 4 or later

Running Btrieve 6.1*x* in a NetWare 3.11 or 3.12 environment requires the AFTER311.NLM, which Btrieve loads automatically.

☐ Adequate memory at the server to load the Btrieve NLM (approximately 528 KB) and the appropriate communications modules

The memory required for the communications modules varies, depending on the values you specify for the Largest Record Size and Number of Remote Sessions configuration options.

☐ Adequate memory for the Btrieve Requester at each workstation

Memory requirements vary, depending on the parameters you specify when you load the requester. Following are the default requester parameters:

- The DOS Requester requires approximately 29 KB (34 KB for BREQUEST 6.10e).

- The OS/2 Requester requires approximately 40 KB.

- The MS Windows Requester requires approximately 30 KB. (To use the MS Windows Requester, you must also have the DOS Requester loaded.)

If you have an earlier version of NetWare Btrieve loaded on your system, unload it before you install NetWare Btrieve 6.1*x*, as described in "Unloading an Earlier Version of NetWare Btrieve" on page 51.

NetWare Btrieve 6.1*x* Program Files

This section provides information about the files that comprise NetWare Btrieve. The 6.1*x* program files can be categorized as follows:

- Files Related to the NetWare 3 Environment
- File Related to the NetWare 4 Environment
- Files Related to Server Operations
- Files Related to DOS Workstation Operations
- Files Related to MS Windows Workstation Operations

Following are brief definitions of the files installed by the NetWare INSTALL utility (described on page 58). Some file, such as the utilities, the requesters, and the communications-related files, are discussed in more detail in this and subsequent chapters.

Files Related to the NetWare 3 Environment

The following files are related to Btrieve operations carried out in a NetWare 3 environment:

- AFTER311.NLM—NLM that NetWare Btrieve v6.1*x* requires in a NetWare 3 environment.
- CLIB.NLM—C Library for the NetWare 3 environment.
- NWLOCALE.DLL—National Language Support DLL for the NetWare 3 environment. The /PUBLIC/WIN and /PUBLIC/OS2 directories each contain a different NWLOCALE.DLL.
- NWSNUT.NLM—Graphics User Interface (GUI) NLM for the NetWare 3 environment.

File Related to the NetWare 4 Environment

The following file is related to Btrieve operations performed in a NetWare 4 environment:

- CLIB.NLM—C Library for the NetWare 4 environment.

Files Related to Server Operations

The following files are related to Btrieve operations performed on a NetWare server:

- BDIRECT.NLM—Directory Services utility that allows you to install or remove a Btrieve server object in a NetWare 4 environment.
- BDIRECT.MSG—Message file for BDIRECT.NLM (for the NetWare 4 environment).
- BDROUTER.NLM—NLM that provides Directory Services support in a NetWare 4 environment.
- BDROUTER.MSG—Message file for BDROUTER.NLM (for the NetWare 4 environment).
- BREBUILD.NLM—Rebuild utility that lets you convert existing NetWare Btrieve v5.x files to NetWare Btrieve v6.1 format.
- BREBUILD.MSG—Message file for BREBUILD.NLM.
- BROUTER.NLM—NLM that allows server-based NetWare Btrieve applications to access Btrieve databases on remote servers.
- BROUTER.MSG—Message file for BROUTER.NLM.
- BSETUP.NLM—Btrieve installation and configuration utility.
- BSETUP.MSG—Message file for BSETUP.NLM.
- BSETUP.HLP—Help file for BSETUP.NLM.

- BSPXCOM.NLM—SPX communications agent.

- BSPXCOM.MSG—Message file for BSPXCOM.NLM.

- BSPXSTUB.NLM —NLM that allows you to use the Btrieve Monitor utility (BTRMON.NLM) when BSPXCOM.NLM is not loaded.

- BSPXSTUB.MSG—Message file for BSPXSTUB.NLM.

- BSTART.NCF—Batch file executed to start NetWare Btrieve. This file also loads BSPXCOM.NLM and BROUTER.NLM (if NetWare Btrieve is configured to do so).

- BSTOP.NCF—Batch file executed to stop NetWare Btrieve. This file also unloads BSPXCOM.NLM and BROUTER.NLM (if they are loaded).

- BTRIEVE.NLM—NetWare Btrieve Record Manager (database engine).

- BTRIEVE.MSG—Message file for BTRIEVE.NLM.

- BTRMON.NLM—Btrieve Monitor utility that lets you monitor the activity of Btrieve files, users, and communications resources.

- BTRMON.MSG—Message file for BTRMON.NLM.

- BTRMON.HLP—Help file for BTRMON.NLM.

- BTRVSTUB.NLM—NLM that BROUTER.NLM requires and that provides NetWare Btrieve support for NLMs that need to run in the IOEngine of a NetWare SFT III server.

- BUTIL.NLM—Btrieve Maintenance utility, which is a command line utility that allows you to create, manipulate, and recover Btrieve data files.

- BUTIL.MSG—Message file for BUTIL.NLM.

- RSPXSTUB.NLM—Module that resolves external references for the Btrieve Monitor utility (BTRMON.NLM) when BSPXCOM.NLM is not loaded. If you want to use BTRMON to monitor outgoing requests generated by the NetWare Btrieve Message Router (BROUTER.NLM) to another server and you do not want to load BSPXCOM.NLM, load RSPXSTUB.NLM (instead of BSPXSTUB.NLM) at the server.

- RSPXSTUB.MSG—Message file for RSPXSTUB.NLM.

Files Related to DOS Workstation Operations

The following files are related to Btrieve operations performed on a DOS workstation:

- BREQUEST.EXE—Btrieve Requester for DOS.

- BREQUEST.MSG—Message file for BREQUEST.EXE.

- BREQUTIL.EXE—Btrieve Requester utility that allows you to stop the Requester and to obtain the Requester's version number at a DOS workstation.

- BREQUTIL.MSG—Message file for BREQUTIL.EXE.

- BROLLFWD.EXE—Roll Forward utility for the DOS operating environment. This utility recovers changes made to a Btrieve file between the time of the last backup and a system failure.

- BROLLFWD.MSG—Message file for BROLLFWD.EXE.

Files Related to MS Windows Workstation Operations

The following files are related to Btrieve operations performed on an MS Windows workstation:

- WBROLL.EXE—Roll Forward utility for the Windows operating environment. This utility recovers changes made to a Btrieve file between the time of the last backup and a system failure.

- WBROLLRS.DLL—Resource file for WBROLL.EXE.

- WBTRCALL.DLL—Btrieve Requester for MS Windows.

- WBTRVRES.DLL—Resource file for the Btrieve MS Windows Requester.

- WNDBCNVT.EXE—MS Windows Conversion utility that converts the client-based Btrieve DLL (WBTRCALL.DLL) to WBTRLOCL.DLL.

- NOVDB.INI—Btrieve initialization file for the Btrieve MS Windows Requester.

Files Related to OS/2 Workstation Operations

The following files are related to Btrieve operations performed on an OS/2 workstation:

- PBROLL.EXE—Roll Forward utility for the OS/2 operating environment. This utility recovers changes made to a Btrieve file between the time of the last backup and a system failure.

- PBTRVRES.DLL—Resource file for PBROLL.EXE.

- BTRCALLS.DLL—Btrieve Requester for OS/2.

- NDBCOMM.DLL—Communications handler for the Btrieve Requester for OS/2.

- NDBCNVT.EXE—OS/2 Conversion utility that converts the client-based Btrieve DLL (BTRCALLS.DLL) to BTRLOCL.DLL.

Using the NetWare INSTALL Utility

The NetWare INSTALL utility lets you load the Btrieve program files automatically at the server. For more information on using the NetWare INSTALL utility, refer to your NetWare documentation.

Warning

In a NetWare 4 environment, NetWare Btrieve 6.1*x* runs at the highest privilege level (domain 0), with direct access to the NetWare operating system. You cannot run NetWare Btrieve 6.1*x* in protected mode (domain 1, 2, or 3).

After loading NetWare Btrieve with the INSTALL utility, you can perform one or all of the following procedures, depending on the needs of your installation:

- Ensure that you have uncorrupted Btrieve (data) files using the instructions in “Detecting and Repairing PAT and File Corruption” on page 59.

  Important

  You should perform the recovery procedure immediately after installing NetWare Btrieve 6.1*x* and before proceeding with configuration or your normal operations.

- Configure NetWare Btrieve according to your Btrieve-based application’s requirements, using the guidelines described in “Configuring NetWare Btrieve” on page 63.

- Activate NetWare Btrieve, as discussed in “Starting and Stopping NetWare Btrieve” on page 75.

- Convert your existing Btrieve 5.x files to the 6.x file format, as discussed in “Rebuilding Existing NetWare Btrieve Files” on page 76.

- Register NetWare Btrieve with NDS, as described in “Registering NetWare Btrieve with the Novell Directory Services” on page 93.

- Set up NetWare Btrieve to support NLMs that need to run in the IOEngine of a NetWare SFT III server, following the procedure presented in “Using NetWare Btrieve with a NetWare SFT III Server” on page 97.

- Set up NetWare Btrieve for use with the NetWare Runtime operating system, as explained in “Using NetWare Btrieve with NetWare Runtime” on page 98.

Detecting and Repairing PAT and File Corruption

If you have a previous version of NetWare Btrieve and you are upgrading to 6.1*x*, ensure that you have uncorrupted Btrieve (data) files before continuing with your regular operations.

To detect file corruption, use the Btrieve Maintenance utility (BUTIL.NLM) commands to perform either of the following procedures on *each* Btrieve file:

- Detecting and Repairing PAT Corruption Only—Follow this procedure if you have large files and are interested in detecting only Page Allocation Table (PAT) corruption.

- Detecting and Repairing PAT Corruption and Other Types of File Corruption—Follow this procedure if you are interested in detecting both PAT and other types of file corruption.

Note

For detailed information about using the BUTIL commands, see the section "NetWare Btrieve Maintenance Utility" on page 138 in Chapter 5, "Using Btrieve Utilities."

Detecting and Repairing PAT Corruption Only

Follow these steps to determine whether the PAT pages in your Btrieve files are corrupted and to repair any such corruption:

Procedure

1. **If you do not already have current backups, make backup copies of your existing Btrieve files.**

2. **Determine the page size of your Btrieve files using the following Maintenance utility command on *each file*:**

   ```
   load butil -stat <full_pathname_of_Btrv_file>
   ```

   Note

   The Maintenance utility runs as an NLM, and the NLM environment does not recognize drive letters or environment variables. Therefore, for commands such as STAT that require a filename, that name must include the full pathname, such as SYS:\BTRIEVE\DATA\ACCTNOS.BTR. If you do not specify a volume, the utility assumes that SYS: is the volume name.

   Please keep a record of the size of each file on which you execute the STAT command. You will need to provide it to the utility in step 4.

3. **Use the following Maintenance utility command on *each* Btrieve file to detect PAT corruption:**

   ```
   butil -salvage <full_pathname_of_Btrv_file>
   ```

   Note

   The SALVAGE command does *not* recover records from your Btrieve file. Therefore, this repair procedure does not take as long to complete as it would if you were using the BUTIL -RECOVER command.

   The SALVAGE command examines the records in the file's Page Allocation Table to determine whether corruption has occurred. (The PAT maintains a map of the physical location of each page in the Btrieve file.)

   If the utility does not report PAT corruption, the file is fine. You do not have to take further corrective measures.

   If the utility reports PAT corruption, proceed to step 4.

4. **The utility asks if you wish to repair the file. Select Yes and enter the page size that you recorded for the file in step 2.**

5. **Use the BUTIL -SALVAGE command a second time on each Btrieve file.**

   If the utility fully recovers the PAT pages on the first run, it reports no errors. The file is fine. You do not need to take further corrective measures.

   If the utility again reports PAT corruption, proceed to step 6.

6. **Make sure you have enough disk space to hold two copies of the Btrieve file you are working with.**

7. **Use the following Maintenance utility command to clone the corrupted Btrieve file:**

   ```
   butil -clone <newBtrvFile> <corruptedBtrvFile>
   ```

   The CLONE command creates a new, empty Btrieve file with the same file specifications as the corrupted file, including any indexes but excluding the owner name. The new Btrieve file contains all the defined key characteristics (such as key position, key length, duplicate key values, an so on) included in the original, corrupted file.

8. **Use the following Maintenance utility command to recover the records from the corrupted file and insert them into the new copy of the Btrieve file:**

```
butil -copy <corruptedBtrvFile> <newBtrvFile>
```

The COPY command copies the contents of the corrupted file into the cloned file, retrieving each record from the original file and inserting it into the new file. (The record size must be the same in both files.) After copying the records, COPY displays the total number of records inserted into the new file.

You may now resume normal operations with your NetWare Btrieve application.

Detecting and Repairing PAT Corruption and Other Types of File Corruption

Follow these steps to determine whether the PAT pages in your files or the Btrieve files themselves are corrupted and to repair any corruptions:

1. **If you do not already have current backups, make backup copies of your existing Btrieve files.**

2. **Use the following Maintenance utility command on *each* Btrieve file to detect PAT corruption:**

   ```
   load butil -salvage <full_pathname_of_Btrv_file>
   ```

   The SALVAGE command does *not* recover records from your Btrieve file. Therefore, this repair procedure does not take as long to complete as it would if you were using the BUTIL -RECOVER command.

   The SALVAGE command examines the records in the file’s Page Allocation Table to determine whether corruption has occurred. (The PAT maintains a map of the physical location of each page in the Btrieve file.)

   If the utility does not report PAT corruption, the file is fine. You do not need to take further corrective measures.

   If the utility reports PAT page corruption, proceed to step 3.

3. **The utility asks if you wish to repair the file. Select No.**

4. **Make sure you have enough disk space to hold two copies of the Btrieve file you are working with.**

5. **Create a description file for each Btrieve file that must be recovered.**

See Appendix B, “Description Files,” for instructions on how to create a description file.

6. **Use the following Maintenance utility command to create a duplicate empty Btrieve file:**

   ```
   butil -create <BtrvFile descriptionFile>
   ```

   The CREATE command generates an empty Btrieve file using the file and key characteristics you have specified in the description file.

   Note

   Using the CREATE command ensures an uncorrupted Btrieve header. You can also use the BUTIL -CLONE command to create the duplicate empty Btrieve file, but that command does not guarantee a clean header.

7. **Use the following Maintenance utility command to recover the records from the corrupted file and insert them into the newly created file:**

   ```
   butil -copy <corruptedBtrvFile> <newBtrvFile>
   ```

   The COPY command copies the contents of the corrupted file into the newly created file, retrieving each record from the original file and inserting it into the new file. (The record size must be the same in both files.) After copying the records, COPY displays the total number of records inserted into the new file.

You may now resume normal operation with your NetWare Btrieve application.

Configuring NetWare Btrieve

You can configure Btrieve by setting configuration options. When you load Btrieve with the INSTALL utility, these options are set to their default values (shown in Table 1).

However, your Btrieve application may require different values for these options. To determine which values your application requires, refer to the documentation included with that application.

Table 1
Default Values for Configuration Options

Configuration Option	Default Value
Number of Open Files	20 files
Number of Handles	60 handles
Number of Locks	20 per client
Number of Transactions	15 transactions
Largest Compressed Record Size	8 KB
Largest Record Size	8,192 bytes
Largest Page Size	4,096 bytes
Number of Remote Sessions	15 sessions
Cache Allocation	512 KB
Perform Index Balancing	No
Logging of Selected Files	No
Create Btrieve Files in Pre v6.x Format	No
Create Files as Transactional	No
Configure BSTART.NCF to Load BROUTER	No

Important

BROUTER.NLM is the default Message Router that is loaded through the NetWare INSTALL utility. If you want Directory Services support, you need to unload BROUTER.NLM and load BDROUTER.NLM.

The following section discusses each configuration option individually. You can change the values for the configuration options by running the Btrieve Setup utility (as described in “Running the Setup Utility” on page 71). The Setup utility stores your changes in the BSTART.NCF NetWare command file.

Configuration Options

This section lists the configuration options in the order in which they appear in the Setup utility. For each option, the discussion includes the following:

- Range of acceptable values
- Default value
- Approximate memory required
- Description of the option

Number of Open Files

Range: 1 through 64,000 files
Default: 20 files
Approximate Memory Required: 425 bytes per file

This option specifies the maximum number of unique Btrieve files, including data dictionary (.DDF) files, that can be open at one time at the server.

The value you specify determines the size of the internal tables used to track active files. Each unique Btrieve file on the server counts as one file.

Number of Handles

Range: 1 through 64,000 file handles
Default: 60 handles
Approximate Memory Required: 200 bytes per handle

This option specifies the maximum number of file handles that the Btrieve NLM can use at one time.

Keep in mind that the number of handles is different from the number of open files. That is, if two sessions open the same file on the same server, Btrieve counts it as one open file, but two different file handles. See “Number of Locks” (next) for a definition of *sessions*.

Note

To calculate the maximum number of handles your system requires when accessing files on a workstation, multiply the value you specify for the Number of Open Files option by the maximum number of clients (that is, workstation tasks) that can access Btrieve at one time. (Each Scalable SQL client is also a Btrieve client. That is, each concurrent Scalable SQL request has an associated Btrieve client.)

Number of Locks

Range: 0 through 64,000 locks

Default: 20 locks per client session

Approximate Memory Required: 20 bytes per lock

This option sets the maximum number of records a client session can lock at the server at one time.

Note

A *session* occurs when a client uses the Btrieve Requester or Message Router to communicate with the Btrieve NLM, *or* when an NLM application calls Btrieve directly.

This maximum number of locks applies to whichever type of read lock (single record or multiple record) the client session is using. Single-record locks allow an application to lock only one record at a time. Multiple-record locks allow an application to lock more than one record at a time.

Number of Transactions

Range: 0 through 64,000 transactions

Default: 15 transactions

Approximate Memory Required: 20 bytes + (2 * maximum number of files)

This option sets the maximum number of Btrieve clients that can simultaneously have active transactions at the server. (Each of these clients can have only one active transaction at the server.)

For example, if you specify 6 transactions, Btrieve creates a transaction file at the server (BTRIEVE.TRN in the SYS:SYSTEM directory) and allows a maximum of 6 clients to have one active transaction each at the server.

If you specify a value of 0, no clients can perform a Btrieve transaction.

Largest Compressed Record Size

Range: 0 through 64 KB

Default: 8 KB

Approximate Memory Required: 2,048 bytes * specified value

This option allows you to allocate memory for a compression buffer that Btrieve can use when you access records in a Btrieve file created with the Data Compression file attribute enabled.

Btrieve allocates a compression buffer with a size of 2,048 bytes multiplied by the value you specify for this option. (This specified value can be determined by either the record size or the page size.)

Use the following guidelines when specifying the value for this option:

- If you use compressed Btrieve files, determine the size (in bytes) of the largest record in any of your compressed files. Round the record size to the next whole kilobyte.

  For example, if the largest record you need to access is 1,800 bytes, specify 2 for this option. Btrieve will allocate 4,096 bytes (that is, 2,048 * 2) of memory for the compression buffer.

- If every compressed file you use has Variable-tail Allocation Tables (VATs), set this option to the file's largest page size (in bytes) divided by 128.

  For example, if the largest page size is 1,024 bytes, specify 8 for this option. Btrieve will allocate 16,384 bytes (that is, 2,048 * 8) of memory for the compression buffer.

- If you do not use compressed files, set this value to 0. You cannot improve performance, and may waste memory, by specifying a value higher than you need.

Largest Record Size

Range: 600 through 64,000 bytes

Default: 8,192 bytes

Approximate Memory Required: `recordLength` * (`remoteSessions` /5) + 1 + (`remoteSessions` * `recordLength` / 580) * 600

`recordLength`	Largest record size + 538 bytes
`remoteSessions`	Number of remote sessions

This option specifies the length of the largest record or record chunk that any remote Btrieve application (excluding other NLMs that call Btrieve, such as Scalable SQL) can access at the server.

A record chunk is any arbitrary portion of a record, specified by its offset and length. Specify the length of the record (or record chunk) in bytes. Specifying a value higher than you need does not improve performance and may waste memory.

Note For applications running on workstations, the maximum record length is 57,000 for the DOS Requester, 65,000 for the OS/2 Requester, and 57,000 for the MS Windows Requester.

Largest Page Size

Range: 512 through 4,096 bytes

Default: 4,096 bytes

Approximate Memory Required: Not applicable

This option specifies the maximum page size (in bytes) of any Btrieve file you want to access. The page size must be an even multiple of 512 bytes, but no greater than 4,096 bytes.

Setting the page size to 512, 1,024, 2,048, or 4,096 can improve performance because these page sizes correspond to disk block sizes. However, if you set the page size to 1,536, 2,560, 3,072, or 3,584, a given disk read may span two disk blocks and therefore require two disk accesses per page.

Note NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

Number of Remote Sessions

Range: 1 through 64,000 sessions

Default: 15 sessions

Approximate Memory Required: 150 bytes * number of remote sessions

This option specifies the maximum number of SPX sessions that can access the remote Btrieve NLM at one time. (A session occurs when a client uses the Btrieve Requester or the Message Router to communicate with the remote Btrieve NLM.)

Each session is allocated two packet buffers for Btrieve requests.

Note

If you receive a Status Code 96 (A communications environment error occurred), increase the value for this option. However, do not specify a value higher than you need. Specifying a value that is too high does not improve performance; instead, it uses memory that NetWare or other processes might need.

Cache Allocation

Range: 32 through 64,000 KB

Default: 512 KB

Approximate Memory Required: Not applicable

This option specifies the size of the cache (in kilobytes) that Btrieve allocates.

To achieve best performance, allocate a cache size equal to the sum of the sizes of the files you are using. However, be careful not to take all available cache, especially if the server is running other applications. You cannot improve performance, and may waste memory, by specifying a value higher than you need.

Perform Index Balancing

Range: Yes or No

Default: No

Approximate Memory Required: Not applicable

When an index page becomes full, Btrieve automatically creates a new index page and splits the values in the full index page between the two index pages. The Perform Index Balancing option lets you avoid creating a new index page every time an old one fills up.

If you specify Yes for this option, Btrieve looks for available space in sibling index pages each time an index page becomes full. Btrieve then rotates values from the full index page into the pages that have space available.

Index balancing increases key page utilization, results in fewer index pages, and produces an even distribution of keys among nodes on the same level, thus increasing performance during Get operations.

However, when you specify Yes for this option, Btrieve requires extra time to examine more index pages and may require more disk I/O during insert, update, and delete operations.

Note

You can also specify index balancing on a file-by-file basis by setting a bit in the file flag's word when the file is created. If you specify Yes to the Perform Index Balancing option, Btrieve performs index balancing on every file regardless of the balanced file flag specification.

Logging of Selected Files

Range: Yes or No

Default: No

Approximate Memory Required: Not applicable

This option controls whether Btrieve logs operations executed on selected files.

If you specify Yes, Btrieve logs all operations that change any file listed in the BLOG.CFG file on Btrieve's server volume. If you specify No, Btrieve performs no logging.

Note You specify the files you want Btrieve to log by adding entries to a \BLOG\BLOG.CFG file that you create on the drive that contains the files. If your files are on multiple drives, create a BLOG.CFG file for each drive. The entries in the log file can be in either pre-6.*x* or 6.*x* file format.

For more information about logging and using the Roll Forward utility, see “NetWare Btrieve Roll Forward Utility” on page 171 in Chapter 5, “Using NetWare Btrieve Utilities.”

Create Btrieve Files in Pre v6.*x* Format

Range: Yes or No

Default: No

Approximate Memory Required: Not applicable

This option specifies that all files be created in a Btrieve version prior to 6.*x*.

Use this option only if you need backward compatibility with a previous version of Btrieve. For example, if you want to create files with Btrieve 6.*x* and you want to use those files with Btrieve 5.*x*, specify Yes for this option.

Note Btrieve 6.*x* can read files that previous versions of Btrieve created. In addition, it can write to files using the existing file format. In other words, it writes to 5.*x* files using the 5.*x* format and writes to 6.*x* files using the 6.*x* format.

Create Files as Transactional

Range: Yes or No

Default: No

Approximate Memory Required: Not applicable

This option controls whether Btrieve automatically flags each file as transactional when you create it.

The transactional flag indicates that the NetWare Transaction Tracking System™ (TTS™) is protecting the file. TTS ensures that, when a file is being modified, either all changes are made or no changes are made, thus preventing data damage.

Configure BSTART.NCF to Load BROUTER

Range: Yes or No

Default: No

Approximate Memory Required: Not applicable

This option controls whether the Message Router is loaded during the execution of the BSTART.NCF NetWare command file.

The Message Router allows other NLMs (such as Scalable SQL) to communicate with remote servers on which Btrieve is loaded. If you want to access Btrieve data on a remote server, specify Yes for this option.

Note

BROUTER.NLM is the default Message Router that is loaded through the NetWare INSTALL utility. If you want Directory Services support, you need to unload BROUTER.NLM and load BDROUTER.NLM.

Running the Setup Utility

Use the Setup utility to set the NetWare Btrieve configuration options interactively. The utility automatically checks to see if the values you enter are within the correct ranges.

You can run the Setup utility either at the server console or at a workstation running the NetWare Remote Console utility (RCONSOLE). To run the Setup utility, complete the following steps:

Note

To get help with the Setup utility, press F1. The help information is context sensitive, relevant to your location in the program. To exit the utility, press the Esc key.

Procedure

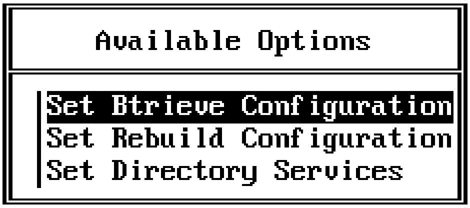
1. **Enter the following command at the prompt to start the utility:**

   ```
   load bsetup
   ```

   Note

   If your SYS:SYSTEM directory does not contain the BSTART.NCF command file, the Setup utility displays a window asking if you want to create it. If that window appears on your screen, create the BSTART.NCF file at this time.

The Available Options menu appears.



This menu provides the following options:

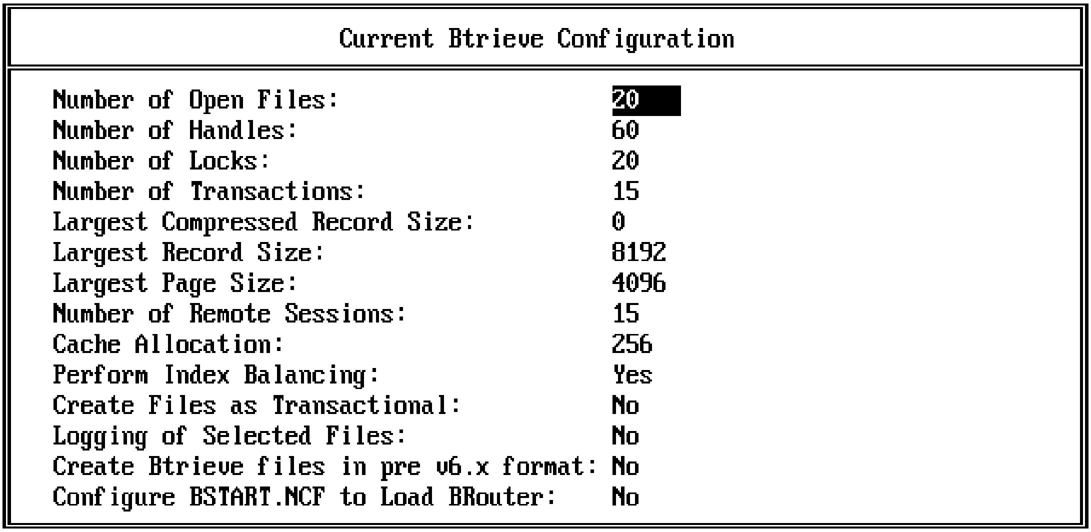
- Set Btrieve Configuration—Use this option to view or modify the current configuration options for NetWare Btrieve, as described in this procedure.
- Set Rebuild Configuration—Use this option to run the Rebuild utility, view or modify the current Rebuild configuration options, and view the Rebuild log file. ("Rebuilding Existing NetWare Btrieve Files" on page 76 discusses the Rebuild utility and its configuration options.)
- Set Directory Services—Use this option to install or remove a Btrieve Server object from the Directory. ("Registering NetWare Btrieve with the Novell Directory Services" on page 93 discusses using the NDS utility.)

2. **To change configuration options, select the first available option, Set Btrieve Configuration.**

   The Current Btrieve Configuration screen, shown in Figure 8, appears.

Figure 8

Setup: Current Btrieve Configuration Screen



```
                    Current Btrieve Configuration

  Number of Open Files:                   20
  Number of Handles:                      60
  Number of Locks:                        20
  Number of Transactions:                 15
  Largest Compressed Record Size:         0
  Largest Record Size:                    8192
  Largest Page Size:                      4096
  Number of Remote Sessions:              15
  Cache Allocation:                       256
  Perform Index Balancing:                Yes
  Create Files as Transactional:          No
  Logging of Selected Files:              No
  Create Btrieve files in pre v6.x format: No
  Configure BSTART.NCF to Load BRouter:   No
```

3. **Use the Up- and Down-Arrow keys to highlight the configuration option you want to change, and press the Enter key.**

   A flashing cursor, shown in Figure 8 highlighting the first configuration option (Number of Open Files), appears.

4. **Use the Backspace key to remove the old value, type the new value, and press Enter.**

   If you enter an invalid value for an option, the utility displays a message indicating the valid values.

   Note

   To see the valid values for any option, move the cursor to that option, press Enter, and then press F1. Refer to “Configuration Options” on page 64 for more information about these options and their values.

5. **Repeat Steps 3 and 4 until you have changed all the options you want to modify.**

6. **When you are ready to exit the Current Btrieve Configuration screen, press the Esc key.**

7. **When the Save Configuration Changes? screen appears, select Yes to save your changes or No to back out of the screen without changing the values.**

   If you select Yes, the Setup utility writes the configuration values you specified to the BSTART.NCF file.

   Important

   Do not edit the BSTART.NCF command file for Btrieve. Always use the Setup utility to make any modifications to the configuration values in the BSTART.NCF file.

   Before your configuration changes can take effect, you must unload the existing Btrieve NLM, BSPXCOM, Message Router, and BDIRECT (using the BSTOP command, described in the following section), and then reload them by entering the BSTART command.

   BSTOP.NCF does not unload BSPXSTUB, RSPXSTUB, or BDROUTER. To unload these modules, enter the UNLOAD command followed by the module name.

8. **To exit the Setup utility, press the Esc key until the Exit window appears, and then select Yes in responde to the prompt.**

Starting and Stopping NetWare Btrieve

This section discusses starting and stopping NetWare Btrieve.

Starting NetWare Btrieve 6.*x*

After installing Btrieve 6.1*x* (as discussed in “Using the NetWare INSTALL Utility” on page 58), you can start it by entering the following at the server console or at a workstation running RCONSOLE:

```
bstart
```

If you have created a Btrieve Server object (as described on page 93), the BSTART.NCF command file for Btrieve 6.*x* automatically loads the NDS Support utility (BDIRECT.NLM). This utility checks to see whether a Btrieve Server object exists. If a Btrieve Server object does exist, the utility activates it.

Note The Btrieve Server object is installed at the level of the current directory context, as defined by the server’s start-up script.

After performing these tasks, the NDS Support utility unloads.

Stopping NetWare Btrieve 6.1*x*

To stop Btrieve 6.1*x*, enter the following command at the server console or at a workstation running RCONSOLE:

```
bstop
```

The BSTOP command runs a NetWare command file (BSTOP.NCF) that unloads the following modules in the order shown:

1. BROUTER.NLM
2. BSPXCOM.NLM
3. BTRIEVE.NLM

Note BSTOP.NCF does not unload BSPXSTUB, RSPXSTUB, or BDROUTER. To unload these modules, enter the UNLOAD command followed by the module name.

If you have created a Btrieve Server object, the BSTOP.NCF command file for Btrieve 6.*x* automatically loads the NDS Support utility (BDIRECT.NLM). If an active Btrieve Server object exists, it is deactivated and the NDS Support utility unloads.

Note

Before issuing the NetWare DOWN command on a server that has an active Btrieve Server object, you must first issue the BSTOP command.

Rebuilding Existing NetWare Btrieve Files

If your database contains files created with versions of Btrieve prior to 6.*x*, you may want to upgrade these files to take advantage of the Btrieve 6.*x* features. Although Btrieve 6.*x* works with 5.*x* files, it does not implement the 6.*x* features.

The Btrieve Rebuild utility (BREBUILD.NLM) converts Btrieve data files to the 6.*x* format. You can run this utility from either the server console or a workstation running RCONSOLE.

Use either of the following methods to run the Rebuild utility:

- Running the Rebuild Utility Interactively—When you run the Rebuild utility interactively, it checks the values you enter to ensure they are within the proper ranges.

- Running the Rebuild Utility from the Command Line—When you use this method, the utility checks your entries and displays messages if the values you entered are not within the proper ranges. With this method, you can specify a command file.

The following sections discuss running the utility using both these methods.

Important

Before running the Rebuild utility, make sure you have unloaded your previous version of Btrieve (as described on page 51), started Btrieve 6.1*x* (as described on page 75), and backed up all your data files. Having backup copies ensures against data loss if a power interruption occurs while you are running the utility.

To ensure that your backup is successful, perform any one of the following operations:

- Close all Btrieve files before running the backup utility.
- Use continuous operations.
- Use a backup utility that opens the Btrieve files in exclusive write mode so that other processes can not write to the files. Ensure that the backup utility has exclusive rights to the files.

Running the Rebuild Utility Interactively

You can run the Rebuild utility interactively through the Set Rebuild Configuration option on the Setup utility's Available Options menu.

As the following sections explain, you can use the Set Rebuild Configuration option to perform the following tasks:

- Configuring the Rebuild Utility
- Executing the Rebuild Utility
- Viewing the Rebuild Log File

Configuring the Rebuild Utility

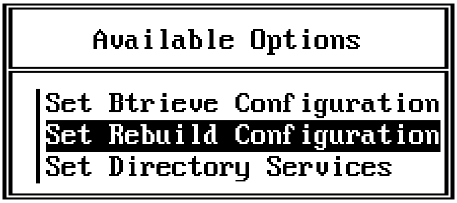
Complete the following steps to set the configuration options that apply to rebuilding your Btrieve files:

Procedure

1. **After starting Btrieve 6.1*x*, load the Setup utility by entering the following command at the prompt:**

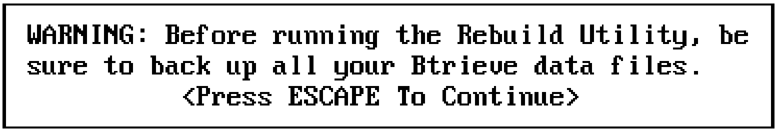
   ```
   load bsetup
   ```

   When the Available Options menu appears, select Set Rebuild Configuration to run the Rebuild utility.

   A warning, shown in Figure 9, appears, indicating you should back up your Btrieve data files before proceeding.

Figure 9
Setup: Backup Warning for Rebuild Option



```
WARNING: Before running the Rebuild Utility, be
sure to back up all your Btrieve data files.
           <Press ESCAPE To Continue>
```

Note

If you do *not* have current backups, press Esc three times. When the Exit Btrieve Setup? prompt appears, select Yes. Create backups and then return to this utility.

2. **Provided you have current backups of your data files, press Esc to advance to the Rebuild Options menu.**

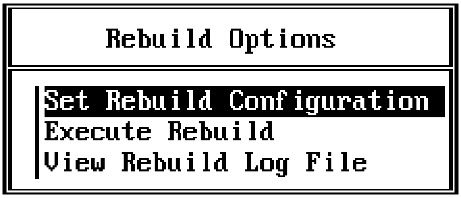
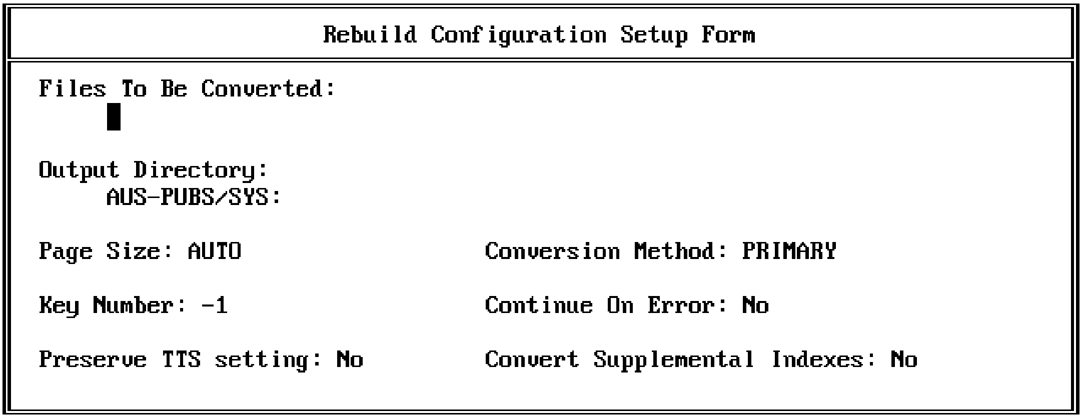


Figure 10
Setup: Rebuild Configuration Setup Form Screen



```
                     Rebuild Configuration Setup Form

 Files To Be Converted:
      █

 Output Directory:
      AUS-PUBS/SYS:

 Page Size: AUTO                  Conversion Method: PRIMARY

 Key Number: -1                   Continue On Error: No

 Preserve TTS setting: No         Convert Supplemental Indexes: No
```

3. **Select the files you want to rebuild, as follows:**

   3a. **With your cursor at the Files To Be Converted entry, press Enter and then press Insert.**

   3b. **When the list of available volumes appears, highlight the volume you want and press Enter.**

   The utility displays the directories available on the selected volume.

   3c. **Highlight the directory you want and press Enter.**

   3d. **Continue highlighting directories (that is, subdirectories) and pressing Enter until you have reached the one that contains the file or files you want to rebuild. Then press Esc.**

   3e. **Choose the files you want to rebuild, as follows:**

   - *To specify more than one file*, enter the filenames manually, using wildcard characters (* or ?).

     For example, you might enter /*.* to specify all files in the selected directory, or you could enter /*.BTR to specify all files in that directory with the extension .BTR.

   - *To specify an individual file*, press Enter to list the filenames in the specified directory. Then highlight the filename you want and press Enter to select it. (The input files that are listed are on the local server.)

4. **At Output Directory, specify the location you want to use for the rebuilt files, as described in the following steps.**

   The default location is the directory you specified for the Files To Be Converted field.

   4a. **Press Enter.**

   4b. **Either type the server or directory name manually and press Enter, or choose from the list of available directories on the current server by entering a valid path and pressing Insert. To select a directory name from the list, highlight the name and press Enter.**

   If you want to store the rebuilt files on a different server, you must type the output server name, volume, and path manually. Then press Enter.

Do not use wildcard characters in the pathname that specifies the location for the rebuilt files.

Note

To store rebuilt files on a different server, Btrieve and the Message Router must be loaded on the server where the original data files reside, and the Btrieve and BSPXCOM NLMs must be loaded on the server that will contain the rebuilt files.

Wherever you store the rebuilt files, you will need enough disk space for the rebuilt files and the temporary files that the utility creates. The utility deletes the temporary files at the end of the conversion process.

4c. After specifying the output directory, use the Down-Arrow key to move to the Page Size field.

5. At Page Size, type the size manually or choose from the list of valid page sizes.

5a. To list the available page sizes, press Enter.

In this list, the AUTO option (the default) means the utility will choose the optimum page size for the files. The EXISTING option means the utility will use the same page size as that of the original files.

5b. To select a size from the list, highlight it and press Enter.

Note

When you use the EXISTING option, the utility changes the page size if the original size will not work. For example, assume you have a Btrieve 5.*x* file with a page size of 1,024 and 24 keys. Since Btrieve 6.*x* supports only 23 keys for a file of that page size, the utility selects a new page size for the file and displays an informative message on the screen.

6. At Key Number, specify a number between 0 and 23 on which to sort the records, or specify –1 to sort the records in physical order. Press Enter.

Important

If you are using Scalable SQL, you *must* specify a key number of 0 when rebuilding the VIEW.DDF file.

7. **At Preserve TTS Setting, specify Y or N to indicate whether you want to preserve the Transaction Tracking System (TTS) bit during conversion. Press Enter.**

   If you specify Y, the utility preserves the bit. If you specify N (the default), the utility clears the bit when creating Btrieve 6.*x* files.

8. **At Conversion Method, select the conversion method as follows:**

   8a. Press Enter.

   8b. Highlight either PRIMARY (the default) or SECONDARY, and then press Enter.

   Important

   If you are using Scalable SQL, you *must* specify the SECONDARY method when rebuilding the VIEW.DDF file. Be aware that the SECONDARY method may create a 6.*x* file in which the records are in a different physical order than in the original 5.*x* file.

   - The PRIMARY method clones the files, drops the indexes, copies the records into the new files, and rebuilds the indexes.

     Since this method is faster and creates smaller files than the SECONDARY method, you should use this method whenever possible. However, if you are using Scalable SQL, you must *not* use this method when rebuilding the VIEW.DDF file.

   - The SECONDARY method clones and copies the files without dropping and replacing indexes. This method may be slower than the PRIMARY method.

9. **At Continue On Error, specify either Y or N and press Enter.**

   If you specify Y, the utility continues if it encounters an error. (The utility notifies you of non-Btrieve files or other errors but continues rebuilding Btrieve files.) If you specify N, the utility stops if it encounters an error and aborts the rebuild process.

10. At Convert Supplemental Indexes, specify Y or N and press Enter.

Important

Do *not* use the Convert Supplemental Indexes option if you access your data files through Scalable SQL.

If you specify Y, the utility converts Btrieve 5.*x* supplemental indexes (which allow duplicates) to Btrieve 6.*x* indexes with linked-duplicatable keys. Btrieve 5.*x* supplemental indexes have, by default, repeating-duplicatable keys.

If you specify N (the default), the utility does not convert the 5.*x* supplemental indexes but preserves them as repeating-duplicatable keys.

11. Press Esc to leave the Rebuild Configuration Setup Form screen.

12. When the utility asks whether to save your changes, select Yes to save them and return to the Rebuild Options menu or No to abandon the changes.

Important

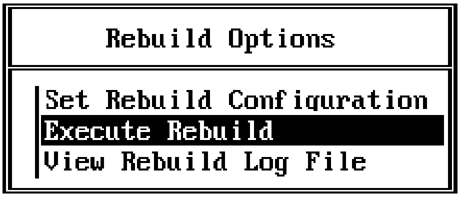
The utility applies the Btrieve 5.*x* file's owner name and level to the Btrieve 6.*x* file.

Executing the Rebuild Utility

After configuring the Rebuild utility, you are ready to rebuild your files.

Procedure

1. **At the Rebuild Options menu, select Execute Rebuild to run the utility.**



   The utility executes and also creates a log file. It then notifies you that the process has completed.

2. **To return to the Rebuild Options menu, press Esc.**

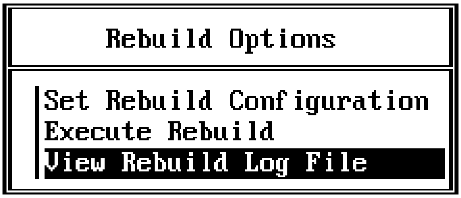
   You can now examine the log file, as discussed in the following section.

Viewing the Rebuild Log File

After rebuilding your files, be sure to check the utility’s log file to see if any errors occurred during the conversion process, as follows:

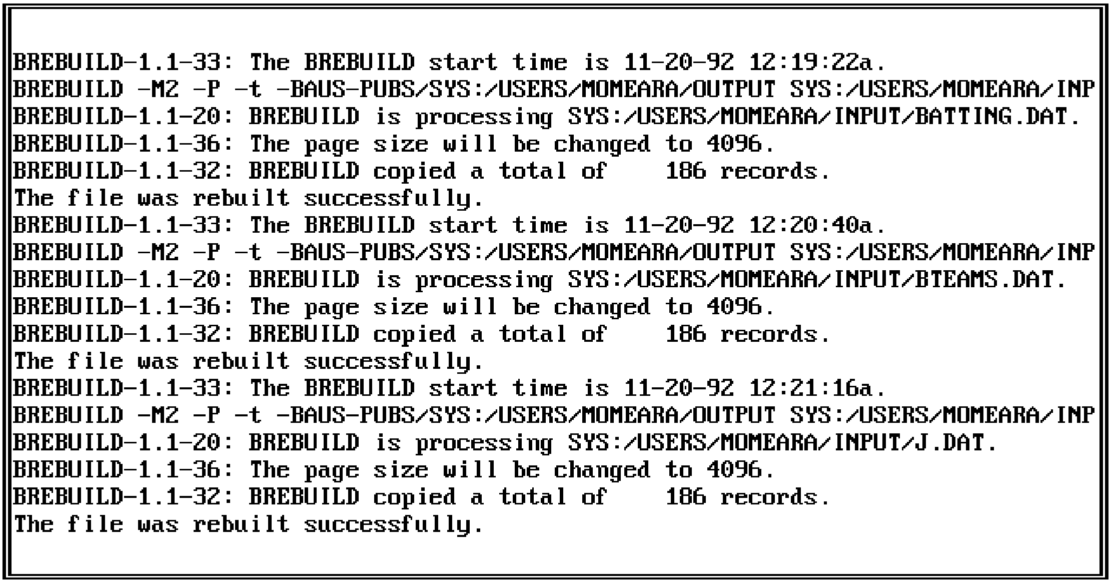
Procedure 23

1. **Select View Rebuild Log File from the Rebuild Options menu.**



The utility displays a log of any errors that occurred while the utility was executing. This log file, REBUILD.LOG, is similar to the example shown in Figure 11.

Figure 11
Setup: Example Rebuild Log File



```
BREBUILD-1.1-33: The BREBUILD start time is 11-20-92 12:19:22a.
BREBUILD -M2 -P -t -BAUS-PUBS/SYS:/USERS/MOMEARA/OUTPUT SYS:/USERS/MOMEARA/INP
BREBUILD-1.1-20: BREBUILD is processing SYS:/USERS/MOMEARA/INPUT/BATTING.DAT.
BREBUILD-1.1-36: The page size will be changed to 4096.
BREBUILD-1.1-32: BREBUILD copied a total of    186 records.
The file was rebuilt successfully.
BREBUILD-1.1-33: The BREBUILD start time is 11-20-92 12:20:40a.
BREBUILD -M2 -P -t -BAUS-PUBS/SYS:/USERS/MOMEARA/OUTPUT SYS:/USERS/MOMEARA/INP
BREBUILD-1.1-20: BREBUILD is processing SYS:/USERS/MOMEARA/INPUT/BTEAMS.DAT.
BREBUILD-1.1-36: The page size will be changed to 4096.
BREBUILD-1.1-32: BREBUILD copied a total of    186 records.
The file was rebuilt successfully.
BREBUILD-1.1-33: The BREBUILD start time is 11-20-92 12:21:16a.
BREBUILD -M2 -P -t -BAUS-PUBS/SYS:/USERS/MOMEARA/OUTPUT SYS:/USERS/MOMEARA/INP
BREBUILD-1.1-20: BREBUILD is processing SYS:/USERS/MOMEARA/INPUT/J.DAT.
BREBUILD-1.1-36: The page size will be changed to 4096.
BREBUILD-1.1-32: BREBUILD copied a total of    186 records.
The file was rebuilt successfully.
```

2. **When you finish viewing the log, press Esc to return to the Rebuild Options menu.**

3. **To exit both the Rebuild utility and the Setup utility, press Esc twice more and specify Yes at the Exit Btrieve Setup? prompt.**

Running the Rebuild Utility from the Command Line

Before running the Rebuild utility from the command line, make sure you have unloaded your previous version of Btrieve (as described on page 51), started Btrieve 6.*x* (as described on page 75), and backed up all your data files. Having a backup copy ensures against data loss if a power interruption occurs while you are running this utility.

Note After rebuilding your files, be sure to check the utility's log file to see if any errors occurred during the conversion. The log file (BREBUILD.LOG) that the Rebuild utility creates is an ASCII text file, which is placed in the SYS:SYSTEM directory. You can examine the log file using a text editor or by running the Rebuild utility interactively and selecting View Rebuild Log File from the Rebuild Options menu (as explained in "Viewing the Rebuild Log File" on page 85).

To run the Rebuild utility from the command line, enter one of the following commands at the prompt:

```
LOAD BREBUILD [–option ...] file
```

or

```
LOAD BREBUILD @commandFile
```

option Specifies the configuration options for the utility.

Precede each option letter with a dash (–). Do not place a space between the dash and the option letter or between the option letter and its value.

You can enter the option letter in uppercase or lowercase.

–B[*path*]

Specifies an alternate location for the rebuilt files. (The default location is the current directory.)

This option lets you rebuild large files on a different volume or on a different server.

To locate the files on a different server, the Btrieve NLM and the Message Router must be loaded on the server where the original data files reside.

The Btrieve and BSPXCOM NLMs must be loaded on the server that will contain the rebuilt files.

Do *not* use wildcard characters in the pathname you specify with the –B option.

–C

Instructs the utility to continue with the next file even if an error occurs.

The utility notifies you of non-Btrieve files or other errors but continues updating Btrieve files.

This option is useful if you have specified wildcard characters for the rebuilt files.

–D

Converts Btrieve 5.*x* supplemental indexes (which allow duplicates) to 6.*x* indexes with linked-duplicatable keys. (By default, the utility preserves the indexes as repeating-duplicatable keys.)

If you access your data files only through Btrieve and your files have a relatively large number of duplicate keys, you can use this option to enhance the performance of the Get Next and Get Previous operations.

Do *not* use the –D option if you access your data files through Scalable SQL.

–M0 | M2

Specifies the conversion method, as follows:

M0

Clones and copies the files without dropping and replacing indexes.

While this method is slower than M2, it is available in case you do not want to rebuild your indexes.

f you are using Scalable SQL, you *must* use the –M0 and –K0 options to rebuild the VIEW.DDF file.

	M2	(Default) Clones the files, drops the indexes, copies the records into the new files, and rebuilds the indexes. Since this method is faster and creates smaller files than the M0 method, you should use it whenever possible. The M2 method may create a 6.x file in which the records are in a different physical order than in the original 5.x file.
–P[*nnn*]	Specifies the page size (in bytes) of the new files. I f you specify –P with no page size, the utility chooses the optimum page size for your file. For example, assume you have a Btrieve 5.*x* file with a page size of 1,024 and 24 keys. Since Btrieve 6.*x* supports only 23 keys for a page size of 1,024, the utility automatically selects a new page size for the file and displays an informative message on the screen. If you do not specify the –P parameter, the utility changes the page size if the original size does not work	

–K[*nn*]

Specifies the key by which the utility reads when rebuilding a file. If you do not specify this option, the utility reads the file in physical order.

You *must* use the –K0 option when rebuilding the Scalable SQL VIEW.DDF file.

–T

Does not preserve the Transaction Tracking System (TTS) bit during conversion.

If you specify this option, the utility clears the TTS bit (if set) when converting a Btrieve 5.*x* file to a Btrieve 6.*x* file.

If you do not specify this option, the utility sets the TTS bit when creating the Btrieve 6.*x* file *if* the 5.*x* file had the TTS bit set.

file

Specifies the set of files to convert.

Use full directory names, including the volume name. You may use wildcards (* and ?) in these filenames.

The Rebuild utility applies the Btrieve 5.*x* file’s owner name and level to the Btrieve 6.*x* file.

@commandFile

Specifies a command file for the utility to execute. You can include multiple entries in one command file.

Each entry in the command file contains the utility options (if any) and the set of files to convert, followed by <end> or [end].

When specifying the files to convert, be sure to use full directory names, including the volume name. You may use wildcards (* and ?) in these filenames.

The following is an example of a Rebuild utility command file:

```
–C sys:\mydir\*.* <end>
–C –P1024 dta:\dir\*.* <end>
–M0 –K0 sys:\nwsql\*.* <end>
```

Examples of Running the Rebuild Utility from the Command Line

The first example places the rebuilt files on another server:

```
load brebuild –Bserv-temp\sys:\newfiles –C –P4096
 sys:\oldfiles\*.btr
```

The next example places the rebuilt files on the same server, but on a different volume:

```
load brebuild –Bvol2:\btrfiles –C –P4096 –M2
 sys:\btrfiles\*.btr
```

Deleting Temporary Files

By default, the Rebuild utility creates temporary files in the same directory in which the conversion takes place. You therefore need enough disk space in that directory (while the Rebuild utility is running) to accommodate both the original file and the new file.

Note You can specify a different directory for storing these files by using the –B option when running the Rebuild utility from the command line, or by using the Output Directory option on the Setup Form screen when running the utility interactively.

The Rebuild utility deletes the original file after rebuilding it, even if the new file is in a different directory.

Normally, the Rebuild utility automatically deletes temporary files when the conversion is complete. However, if a power failure or other serious interruption occurs, the Rebuild utility may not delete the temporary files. If this occurs, look for filenames such as _T-xxxxx.TMP and delete them.

Registering NetWare Btrieve with the Novell Directory Services

Novell Directory Services™ (NDS™), a NetWare 4 component, implements a system database called the Directory that can be distributed over a network. The Directory replaces the bindery used in previous NetWare releases. To provide compatibility with bindery-based NetWare versions, however, the Directory offers you the option of using bindery emulation.

With NDS you can organize network resources (such as users, network servers, and printers) as *objects* within the Directory. The Directory implements a hierarchical tree structure to store objects and their associated attributes. All objects must belong to an *object class*, which is constructed from the *attribute types*. The attribute types define the categories of information that can be associated with objects.

You must have appropriate access rights to read or modify the information stored in the Directory. For more information about NDS features (objects, naming conventions, access control, and bindery emulation) and how to manage the Directory database, refer to the *Installation and Upgrade* manual provided with NetWare 4.

Btrieve 6.*x* provides the NDS Support utility (BDIRECT.NLM), a utility that allows you to register Btrieve with NDS as an object of the class *Btrieve Server*. The Btrieve Server class identifies the host server name, the type of communications protocol supported, and the object's internet address.

You can run the NDS Support utility interactively, through the Btrieve Setup utility, or from the command line. The following sections describe how to use these two methods.

Important

Before running the NDS Support utility, you must set the bindery context in the STARTUP.NCF file or at the server console. The name path for the bindery context must begin with Organization or Organizational Unit. The following example illustrates using a name path for the bindery context that begins with the Organizational Unit.

```
set bindery context = OU=Testing.O=Novell.C=US
```

Installing or Removing the NDS Support Utility Interactively

You can use the Setup utility to either install NetWare Btrieve as an object in the Directory or to remove an existing Btrieve Server object from the Directory.

Installing NetWare Btrieve as a Server Object in the Directory

Follow these steps to install NetWare Btrieve as an object in the Directory using the Setup utility:

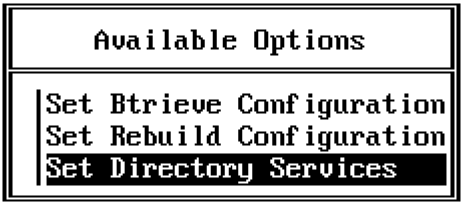
1. **If you are not already in the Setup utility, start it by entering the following command:**

   ```
   load bsetup
   ```

   As indicated previously, the Available Options menu is the first screen you see after starting the Btrieve Setup utility.

Procedure 23

2. **From the Available Options menu, select the Set Directory Services option to install Btrieve as an object in the Directory**

   Available Options

   Set Btrieve Configuration
   Set Rebuild Configuration
   Set Directory Services

   The NDS Support utility determines whether a Btrieve Server object exists for the server on which the utility is loaded. If no Btrieve Server object is found, the utility asks if you want to install one.

3. **Select Yes to install the Btrieve Server object.**

   The utility prompts you for the Directory *object name* (like a username, but called an object name under NDS) and password to be used to log in to the network. The login object name must have the appropriate rights to create or delete objects in the Directory.

4. **Enter the Directory object name and password to be used to log in to the network.**

   If you are successful in installing the Btrieve Server object, the utility modifies the BSTART.NCF file (which implements the LOAD BDIRECT –A command for activation) and notifies you that the installation succeeded.

Note The BSTART.NCF file must reside in the SYS:SYSTEM directory. Installing a Btrieve Server object does not activate it. You must use BSTART (described on page 75) to load Btrieve and activate the Btrieve Server object.

Activation causes the network address attribute of the Btrieve Server object to be filled in with the network address of the server.

Removing an Existing Btrieve Server Object from the Directory

In the case of an existing active Btrieve Server object, the utility notifies you that this object is active and asks if you want to remove it from the Directory. If you select Yes, the NDS utility removes the Btrieve Server object. If you select No, the utility returns you to the previous screen.

If you are successful in removing the Btrieve Server object, the utility modifies the BSTOP.NCF file (which implements the LOAD BDIRECT –D command for deactivation).

Note The BSTOP.NCF file must reside in the SYS:SYSTEM directory. Removing a Btrieve Server object does not deactivate it. You must use BSTOP (described on page 75) to unload Btrieve and deactivate the Btrieve Server object.

Deactivation removes the network address from the attribute of the Btrieve Server object. (If the removal fails, the utility notifies you.)

Running the NDS Support Utility from the Command Line

You can install the NDS Support utility as a standalone NLM.

Procedure

1. **To install Btrieve as an object in the Directory (or to remove an existing Btrieve Server object from the Directory), enter the following command at the server console prompt:**

   ```
   load bdirect
   ```

   The NDS Support utility determines whether a Btrieve Server object exists for the server on which the utility is loaded. If no Btrieve Server object is found, the utility asks if you want to install one.

   In the case of an existing Btrieve Server object, the utility asks if you want to remove it from the Directory.

2. **Specify Yes to install (or remove) the Btrieve Server object.**

   The utility prompts you for the Directory *object name* (like a username but called an object name under NDS) and password that will be used to log in to the network. The login object name must have the appropriate rights to create or delete objects in the Directory.

   If you specify No, the utility terminates.

3. **Enter the Directory object name and password to be used to log in to the network.**

   If you are installing the Btrieve Server object, the utility modifies the BSTART.NCF file (which implements the LOAD BDIRECT –A command for activation). Activation causes the network address attribute of the Btrieve Server object to be filled in with the network address of the server.

   If you are removing the Btrieve Server object, the utility modifies the BSTOP.NCF file (which implements the LOAD BDIRECT –D command for deactivation). Deactivation removes the network address from the attribute of the Btrieve Server object. (If the removal fails, the utility notifies you.)

Note

The BSTART.NCF and BSTOP.NCF files must reside in the SYS:SYSTEM directory. Installing a Btrieve Server object does not activate it. You must use BSTART (described on page 75) to load Btrieve and activate the Btrieve Server object or use BSTOP (described on page 75) to unload Btrieve and deactivate the Btrieve Server object.

Using NetWare Btrieve with a NetWare SFT III Server

An enhanced NetWare Btrieve Message Router (BROUTER.NLM) and a new communications module (BTRVSTUB.NLM) were provided with NetWare Btrieve 6.10b to support NLMs that need to run in the IOEngine of a NetWare SFT III server.

BROUTER.NLM redirects all Btrieve calls from the IOEngine to NetWare Btrieve running in the MSEngine (Mirror Server Engine).

To run NLMs that require Btrieve in the IOEngine, perform the following steps:

Procedure

1. **Load BTRIEVE.NLM and BSPXCOM.NLM in the MSEngine. You can configure the BSTART.NCF NetWare command file to load these NLMs.**

   Depending on the particular NLM that you will be running in the IOEngine, you may need to increase BSPXCOM’s data buffer size with the Btrieve Setup utility (BSETUP.NLM). (See the discussion of the configuration options “Largest Record Size” on page 66 and “Number of Remote Sessions” on page 67.)

2. **Load AFTER311.NLM in the IOEngine.**

3. **Load BTRVSTUB.NLM in the IOEngine. This NLM provides Btrieve API names so that BROUTER.NLM may be loaded.**

4. **Load BROUTER.NLM in the IOEngine.**

   Depending on the particular NLM that you will be running in the IOEngine, you may need to increase BROUTER's data buffer size with the Btrieve Setup utility.

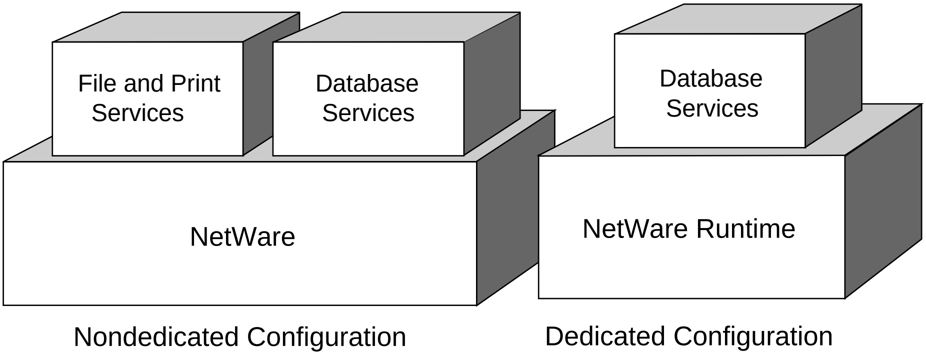
Using NetWare Btrieve with NetWare Runtime

The NetWare Runtime™ serialized NetWare operating system differs from the other versions of NetWare in that it grants file service access to only one NetWare login client connection. This login is for system administration purposes.

NetWare Runtime does not limit the number of SPX or AppleTalk* connections between client applications and NLM-based services. Consequently, NetWare Runtime does not limit the number of users that can access NetWare Btrieve running on the Runtime server.

Figure 12 illustrates the relationship between a NetWare configuration that is not dedicated to database services and a NetWare Runtime configuration that is dedicated to database services.

Figure 12
NetWare Configurations



Reasons to Use NetWare Runtime

Running NetWare Btrieve on a server dedicated to database management ensures that all the server’s processing power is directed toward database applications. If you anticipate heavy file service activity, a dedicated database server makes that activity more efficient because it frees the nondedicated server to devote all its resources to file services.

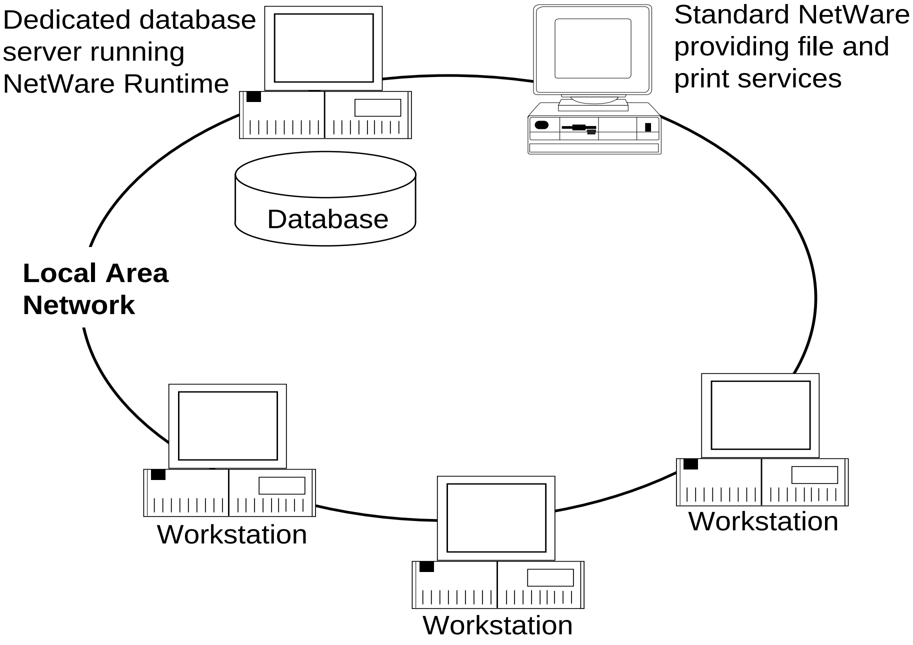
Having a dedicated database server is also particularly effective in preventing slow performance on the network during periods of heavy file service activity. To optimize network performance, you can configure so as to include a dedicated database server in addition to other servers offering full NetWare 4 services.

Figure 13 shows an example installation for NetWare Runtime.

Installing NetWare Runtime

The installation procedure for NetWare Runtime is identical to installing the other versions of NetWare. Refer to the documentation that accompanies your NetWare Runtime software for instructions.

Figure 13
NetWare Runtime Installation


Dedicated database server running NetWare Runtime
Standard NetWare providing file and print services
Database
Local Area Network
Workstation
Workstation
Workstation

Special Notes on NetWare Runtime

NetWare Runtime supplies a facility (NLICLEAR.NLM) that clears unused connections.

NetWare Runtime provides only one available client connection. Even after the administrator logs out of the application server, a connection is maintained between the workstation and the server. If a second administrator wants to log on to the application server from a second workstation, the single connection will be unavailable.

The NLICLEAR facility is important for NetWare Runtime because, at intervals, NLICLEAR automatically clears the unused connections left after an administrator logs out of the application server. This allows another administrator to log in to and administer the database server.

Special Notes on NetWare Btrieve

When you load the Btrieve DOS or OS/2 Requester using the option /C:1, *username*, *password*, Btrieve logs in to the NetWare Runtime server with the specified username and the corresponding password. Btrieve also obtains a temporary connection number, which it uses to distinguish between users.

Btrieve verifies that the user has the acceptable rights to open or create a file. Btrieve then logs out of the server, using the temporary connection number. If the user has the appropriate file access rights, Btrieve continues; otherwise, it returns an error.

Note The administrator must set up file access rights on the NetWare Runtime server.

When you load the Btrieve DOS or OS/2 Requester using the /C:1 default option (without specifying a username and password) and then attempt to read a file on the NetWare Runtime server, the Requester must determine what login username Btrieve can use to maintain NetWare security. Btrieve then uses that username to log in temporarily for the client.

In contrast, if the Requester detects that there is no connection, or if it cannot find a valid login username, the Requester returns an error.

For more information about the NetWare Runtime server support option available with the DOS and OS/2 Requesters, see Chapter 4, “Configuring and Using the Requesters.”

chapter 4

Configuring and Using the Requesters

The Btrieve® record management system provides a DOS Requester, an OS/2 Requester, and an MS Windows Requester. This chapter provides configuration options and instructions for loading and unloading the requester in each operating system environment.

This chapter consists of the following sections:

- “DOS Requester” on page 104
- “OS/2 Requester” on page 110
- “MS Windows Requester” on page 115

Important The Btrieve Requesters have changed (primarily in key definition) between previous versions and Btrieve 6.1*x*. Therefore, all workstations accessing the Btrieve 6.1*x* NLM™ must use the Btrieve 6.1*x* Requesters.

DOS Requester

You must load the Btrieve DOS Requester, BREQUEST.EXE, at a workstation running DOS before that workstation can access network Btrieve files using the Btrieve NLM. The DOS Requester loads into a DOS workstation's memory as a Terminate and Stay Resident (TSR) program.

You can access local as well as remote files by running both client-based (local) Btrieve and the requester at your workstation.

Note Client-based Btrieve is available only as part of the Btrieve Developer's Kit, which is available through Btrieve Technologies Inc. at 1-800-BTRIEVE and must be purchased separately.

DOS Requester Configuration Options

There are five configuration options for the NetWare Btrieve DOS Requester:

- NetWare® Runtime™ Server Support (/C)
- Data Message Length (/D)
- DOS Session Load (/L)
- Real-Time Data Compression (/O)
- Help (/?)

Note Previous versions of the DOS Requester also supported the /S and /R options. The DOS Requester 6.*x* accepts the /S and /R options for backwards compatibility but otherwise ignores them.

NetWare Runtime Server Support (/C)

Range: None

Default: /C:1

Memory Required: Not applicable

This option allows you to enable or disable NetWare Runtime server support.

`/C:0 | /C:1 | /C:1,`*`username`*`,`*`password`*

/C:0 Disables NetWare Runtime server support.

/C:1 Enables NetWare Runtime server support.

Btrieve looks at the username for the drive (current server) on which you are presently running.

If the username is SUPERVISOR, Btrieve searches for another username in the table of usernames for the servers onto which you are logged.

If the username is not SUPERVISOR, Btrieve searches for that username on the NetWare Runtime server. If it is not a valid username, Btrieve returns an error at the time of the Open or Create request.

/C:1,*username*,*password*

Enables NetWare Runtime server support. Btrieve verifies the specified username and password for the NetWare Runtime server, returning an error if the specified username is not found or the password is invalid.

username Preferred login name on the NetWare Runtime server.

If you specify SUPERVISOR for the username, Btrieve returns an error and the DOS Requester will not load.

password Login password for the specified user.

For information on using Btrieve with NetWare Runtime, see “Using NetWare Btrieve with NetWare Runtime” on page 98 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Data Message Length (/D)

Range: 532 through 57,000 bytes

Default: 4,096 bytes

Memory Required: 538 bytes + data message length

This option specifies the length of the largest record (or the largest portion or chunk of a record) you want to access through Btrieve. If you omit this option, the requester uses the default value, 8,192.

The DOS Requester uses this value to calculate the length of the data message buffer reserved for passing records between NetWare Btrieve and your applications. The requester maintains one copy of the data message buffer.

The value you enter here should not exceed the largest record size you configure for NetWare Btrieve through the Setup utility, as that is the maximum message that BSPXCOM can receive. (For more information, refer to the section “Largest Record Size” on page 66 in Chapter 3.)

Specify the data message length in bytes. For example, if the largest record your application uses is 3,000 bytes, specify the /D option as follows:

```
/d:3000
```

Note Specifying a higher value than you need for the /D option does not improve performance and may waste memory.

DOS Session Load (/L)

Range: Not applicable

Default: Not applicable

Memory Required: Not applicable

This option allows you to load another instance of BREQUEST even if one is already loaded. This option is useful if you want to run MS Windows-based Btrieve applications using the DOS Requester while running DOS VM applications that also use the DOS Requester.

To run MS Windows applications that use the DOS Requester, you must load BREQUEST before starting MS Windows. In order to run DOS applications in MS Windows, you must load BREQUEST in *each* DOS session. However, if you load BREQUEST outside MS Windows, you cannot load it again in a DOS session.

For MS Windows applications using the DOS Requester, load BREQUEST outside MS Windows. In each MS Windows DOS session that will be running a Btrieve application, load BREQUEST with the /L option. Doing so loads another instance of BREQUEST that is available only to that DOS session.

This operation provides the DOS session with its own copy of BREQUEST and prevents the DOS session from using the instance of BREQUEST that you loaded before starting MS Windows.

Real-Time Data Compression (/O)

Range: None

Default: No compression

Memory Required: Approximately 32 KB on the workstation and 32 KB per client on the server

In many cases (such as when implementing extended reads or when using VATs), this option can help reduce network traffic by reducing the number of packets required to complete a request to NetWare Btrieve.

Using this option may, however, adversely affect memory and performance. Compressing and decompressing data takes extra CPU time on both the server and client sides. Because of overhead, you should not use this option with fast networks or with slow workstations for clients.

Help (/?)

The /? option lists the other options that are available (/C, /D, and /O) and mentions that although the /S and /R options are accepted for backwards compatibility, Btrieve 6.*x* ignores them.

Loading the DOS Requester

Load the DOS Requester at the workstation by entering the following command:

```
[path] BREQUEST [option]
```

path The pathname to the directory where the DOS Requester is stored.

You can omit the pathname if the DOS Requester is stored on the default drive or if it is located in a directory in your search path.

option Any of the configuration options (/C, /D, /L, /O, or /?).

For example, if the requester is on the default drive and you want to specify a 2,048-byte data message length, enter `the following:`

```
brequest /d:2048.
```

Note The forward slash (/) before the configuration option is the only valid character you can use. If you specify a dash (–) or a backslash (\), the DOS Requester may load improperly.

Unloading the DOS Requester

The DOS Requester utility, BREQUITIL.EXE, has three commands that enable you to perform the following tasks:

- Unload the requester (BREQUITIL -STOP)
- Perform a NetWare Btrieve reset operation (BREQUITIL -RESET)
- Determine the version of the requester (BREQUITIL -VER)

To unload the DOS Requester, enter the following at the workstation where the requester is loaded:

```
brequtil -stop
```

Note

If files have been left open, as happens, for example, when an application does not issue a Close (1) operation for each open file, simply logging out of one or more servers from a workstation does not close Btrieve files or terminate the Btrieve SPX connection to the server. While BREQUTIL -STOP does perform these tasks, it does not necessarily release all the workstation resources.

To release all the resources used by Btrieve at a workstation, you must execute the BREQUITIL -RESET command, which performs a Btrieve Reset (28) operation:

```
brequtil -reset <Connection ID>
```

The RESET command aborts any pending transactions, releases all locks, and closes all open files for the workstation. It does not, however, unload the DOS Requester, so you should first enter the STOP command and then the RESET command.

To determine the version of your DOS Requester, enter the following command:

```
brequtil -ver
```

OS/2 Requester

The following files must be loaded in a directory listed in an OS/2 workstation’s LIBPATH before a Btrieve application can access the Btrieve NLM™ from that workstation:

Checklist

- □ BTRCALLS.DLL—The Btrieve dynamic link Requester for OS/2 workstations
- □ NDBCOMM.DLL—The communications Requester for OS/2 workstations

  NDBCOMM.DLL provides the necessary data communications between the workstation and the Btrieve NLM.

You can access local as well as remote files by running both client-based (local) Btrieve and the OS/2 Requester at your workstation. To do so, you must use the OS/2 Conversion utility (NDBCNVT.EXE) to convert the client-based BTRCALLS.DLL to BTRLOCL.DLL. (By default, the OS/2 Requester and client-based Btrieve have the same name.)

Note

Client-based Btrieve is available only as part of the Btrieve Developer’s Kit, which is available through Btrieve Technologies Inc. at 1-800-BTRIEVE and must be purchased separately.

OS/2 Requester Configuration Options

You do not need to specify any configuration options for the OS/2 Requester. Since the internal tables that control the options are not fixed, the tables will grow as needed.

There are, however, three configuration options for the OS/2 Requester that you can specify manually if you need to change these values for your installation: NetWare Runtime Server Support (/C), Data Message Length (/D), and Number of Servers (/S).

Important

Although you can set the initial size of the tables using the Data Message Length (/D) and the Number of Servers (/S) options, setting table sizes is *not* recommended.

NetWare Runtime Server Support (/C)

Range: None

Default: /C:1

Memory Required: Not applicable

This option allows you to enable or disable NetWare Runtime server support.

`/C:0 | /C:1 | /C:1,`*`username`*`,`*`password`*

/C:0 Disables NetWare Runtime server support.

/C:1 Enables NetWare Runtime server support. Btrieve looks at the username for the drive (current server) on which you are presently running.

If the username is SUPERVISOR, Btrieve searches for another username in the table of usernames for the servers onto which you are logged.

If the username is not SUPERVISOR, Btrieve searches for that username on the NetWare Runtime server.

If it is not a valid username, Btrieve returns an error at the time of the open or create request.

/C:1,*username*,*password*

Enables NetWare Runtime server support.

Btrieve verifies the specified username and password for the NetWare Runtime server. Btrieve returns an error if the specified username is not found or the password is invalid.

username Preferred login name on the NetWare Runtime server.

If you specify SUPERVISOR for the username, Btrieve returns an error and the OS/2 Requester will not load.

password Login password for the specified user.

For information on using Btrieve with NetWare Runtime, see “Using NetWare Btrieve with NetWare Runtime” on page 98 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Data Message Length (/D)

Range: 532 through 65,000 bytes

Default: 4,096 bytes

Memory Required: 538 bytes + data message length

The /D option specifies the length of the largest record (or the largest portion or chunk of a record) you want to access through Btrieve.

The OS/2 Requester uses this value to calculate the length of the data message buffer reserved for passing records between Btrieve and your applications. (The requester maintains one copy of the data message buffer.)

The value you enter here should not exceed the value you specified for the Largest Record Size configuration option in the Setup utility. (For more information, refer to the section “Largest Record Size” on page 66 in Chapter 3.) The value for this option represents the maximum message length that BSPXCOM can receive.

Specify the record length in bytes. For example, if the largest record your application uses is 3,000 bytes, specify the /D option as follows:

```
/d:3000
```

Note Specifying a higher value than you need for the /D option does not improve performance.

Number of Servers (/S)

Range: 1 through 255

Default: 8

Memory Required: 27 bytes per server

The /S option specifies the number of servers on the network on which the NetWare Btrieve NLM is active.

Configuring the OS/2 Requester

Set the OS/2 Requester configuration options using the following command:

```
SET BRQPARMS=option
```

option Any of the three configuration options (/C, /D, or /S).

Do not include a space between BRQPARMS and the equal sign. You can, however, insert a space between each configuration option you specify.

For example, to specify a 10,240-byte data message length and four servers, issue the following command:

```
set brqparms=/d:10240  /s:4
```

Note

The forward slash (/) before the configuration option is the only valid character you can use. If you specify a dash (–) or a backslash (\), the OS/2 Requester may load improperly.

Loading the OS/2 Requester

An application may load the Btrieve for OS/2 Requester in one of the following two ways:

- Implicitly—An application can implicitly load the OS/2 Requester either by linking with the import library, BTRCALLS.LIB, or by specifying imported functions in the application definition file.

  In either case, the operating system loads the OS/2 Requester automatically when the application is started.

- Explicitly—An application can load the OS/2 Requester explicitly using the operating system API functions.

  When the application loads the Requester explicitly, the operating system does not load the OS/2 Requester until notified to do so.

Unloading the OS/2 Requester

At an OS/2 workstation, the operating system removes the dynamic link routines from memory when the application terminates, or when the application explicitly unloads the OS/2 Requester using the operating system API functions.

Note If files have been left open, as happens, for example, when an application does not issue a Close (1) operation for each open file, simply logging out of one or more servers from a workstation does not close Btrieve files or terminate the Btrieve SPX connection to the server. To close files and terminate the connection, you must either unload the OS/2 Requester or reboot the workstation.

MS Windows Requester

In the MS Windows environment, you must load the DOS Requester, BREQUEST.EXE, before starting MS Windows. MS Windows-based Btrieve applications access the requester by means of a DLL, WBTRCALL.DLL, which uses the DOS Protected Mode Interface (DPMI) that MS Windows provides.

The MS Windows Requester (that is, WBTRCALL.DLL) is the Btrieve interface DLL for MS Windows 3.*x*. The DLL provides the same API as client-based Btrieve and requires no modification to the application.

You can access local as well as remote files by running both client-based (local) Btrieve and the MS Windows Requester at your workstation. If you want to run both client-based Btrieve and the requester, you must also run the MS Windows Conversion utility (WNDBCNVT.EXE) to convert the client-based WBTRCALL.DLL to WBTRLOCL.DLL.

Note Client-based Btrieve is available only as part of the Btrieve Developer's Kit, which is available through Btrieve Technologies Inc. at 1-800-BTRIEVE and must be purchased separately.

MS Windows Requester Configuration Options

The following list describes the available configuration options for the MS Windows Requester. These options should be specified in the NOVDB.INI file under `[brequestDPMI]`.

Note NOVDB.INI is the Novell® initialization file for the MS Windows Requester. It is included on the distribution diskette.

Tasks=#	Specifies how many concurrent tasks may use the MS Windows Requester. The range is 1 through 255. The default value is 10.
Local=Yes/No	Instructs the MS Windows Requester to use client-based Btrieve for accessing files locally. The default value is No.

Chkparms=Yes/No

Instructs the MS Windows Requester to validate all Btrieve call parameters passed to it. (Use this option only for debugging during development.) The default value is No.

Free Memory=Yes/No

Allocates and frees real-mode memory on each request.

The MS Windows Requester uses a buffer of real-mode memory to communicate with the DOS Requester. Since real-mode memory is a scarce resource in MS Windows, the application should not retain it long-term. The default value is No.

Important

Specifying Yes to the Free Memory option degrades performance.

Loading the MS Windows Requester

The DOS Requester must be loaded before the MS Windows Requester can load. If you want to run MS Windows-based Btrieve applications, the DOS Requester must be loaded in the DOS environment before you start MS Windows.

To load the DOS Requester, use the WINSTART.BAT file, or enter the following at the DOS prompt before loading MS Windows:

```
brequest
```

An application may load the MS Windows Requester in one of the following ways:

- Implicitly—The application can implicitly load the MS Windows Requester by either linking with the import library (WBTRCALL.LIB,) or by specifying imported functions in the application definition file.

  When an application loads a DLL implicitly, the operating system automatically loads the DLL when the application is started.

- Explicitly—The application may load the MS Windows Requester explicitly using the operating system API functions.

The operating system does not load the DLL until notified to do so.

Important

If you want to run a DOS-based Btrieve application in the DOS box and a MS Windows-based Btrieve application in MS Windows at the same time, you must have the DOS Requester loaded in each DOS session. However, if you have already loaded the DOS Requester before loading MS Windows, you cannot load the DOS Requester in any subsequent DOS session. Consequently, you cannot run the DOS-based Btrieve application in the DOS box.

For MS Windows applications using the DOS Requester, load BREQUEST outside Windows. In each MS Windows DOS session that will be running a Btrieve application, load BREQUEST with the /L option. Doing so loads another instance of BREQUEST that is available only to the DOS session. This operation provides the DOS session with its own copy of BREQUEST and prevents the DOS session from using the instance of BREQUEST that you loaded before starting Windows.

Unloading the MS Windows Requester

At a MS Windows workstation, the operating system removes the dynamic link routines from memory when the application terminates, or when the application explicitly unloads the DLLs using the operating system API functions.

Note

If Btrieve files are left open in an application because the application did not issue a Close (1) operation, simply logging out of one or more servers from a workstation does not close the files or terminate the Btrieve SPX connection to the server. The files remain open until you reboot the workstation, unload the DOS Requester using BREQUTIL –STOP, or execute a reset operation using BREQUTIL -RESET. (Remember, however, that a reset operation does not unload the DOS Requester.)

chapter 5

Using NetWare Btrieve Utilities

This chapter includes information about the following NetWare Btrieve utilities:

- “NetWare Btrieve Monitor Utility” on page 119—This utility monitors the activity of the Btrieve® record management system.

- “NetWare Btrieve Maintenance Utility” on page 138—This utility imports and exports Btrieve data and transfers data from one Btrieve file to another.

- “NetWare Btrieve Roll Forward Utility” on page 171—This utility recovers changes made to a Btrieve file between the time of the last backup and a system failure.

NetWare Btrieve Monitor Utility

The NetWare Btrieve Monitor utility (BTRMON.NLM) allows you to monitor Btrieve activities on a server. It provides information that is useful for both database administration and application programming diagnostics.

Note The information you receive from the Btrieve Monitor utility pertains only to the activity of the NLMs on the *current* server.

The Btrieve Monitor utility runs as an NLM™ at the server. You can access it at the server console or through RCONSOLE at your workstation.

This section discusses the following topics:

- “System Requirements for the NetWare Btrieve Monitor Utility” on page 120

- “Starting the NetWare Btrieve Monitor Utility” on page 121

- “Selecting Options from the Available Options Menu” on page 121

- “Monitoring Btrieve Files with the Active Resources Option” on page 123
- “Monitoring Btrieve Users with the User Information Option” on page 127
- “Monitoring Resources with the Resource Usage Option” on page 132
- “Monitoring SPX Activity with the Communication Statistics Option” on page 135

System Requirements for the NetWare Btrieve Monitor Utility

To run the NetWare Btrieve Monitor utility, be sure that the following software is loaded on the NetWare server you are accessing:

Checklist ✓

□ NetWare® 4 or later

Running NetWare Btrieve 6.1*x* in a NetWare 3.11 or 3.12 environment requires the AFTER311.NLM, which Btrieve loads automatically.

□ Btrieve 6.1*x* or later

□ One of the following communications modules: BSPXCOM.NLM, BSPXSTUB.NLM, or RSPXSTUB.NLM

For information about these NLMs, see Communications Programs on page 29 in Chapter 2, “Btrieve Architecture.”

In addition, make sure the following files are loaded in the SYS:SYSTEM directory of the server (the NetWare INSTALL utility places them there automatically during installation):

□ BTRMON.NLM—NetWare Btrieve 6.1*x* Monitor utility

□ BTRMON.HLP—Help file for the Monitor utility

Starting the NetWare Btrieve Monitor Utility

To start the Btrieve Monitor utility, enter the following command at the server console prompt:

```
load btrmon
```

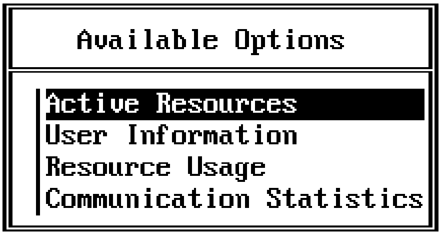
Alternatively, you can include the command line option /R with the LOAD command. The /R option specifies the update time for the statistics screens that the utility dynamically updates. The range of valid values for this option is from 500 to 60,000 milliseconds; the default value is 2,000 milliseconds.

For example, if you want to start the Btrieve Monitor utility and instruct it to update the statistics screens every 3,500 milliseconds, enter the following command:

```
load btrmon /r3500
```

Selecting Options from the Available Options Menu

The Available Options menu is the first screen you see after starting the Btrieve Monitor utility.



You can select one of the following options from this menu:

- Active Resources

  Displays statistics about active Btrieve data files. For more information, see “Listing Active (Open) Files” on page 123.

- User Information

  Displays statistics about the users currently using the Btrieve NLM. This option also allows you to delete a user’s SPX connection. For more information, see “Listing All Active Btrieve Users” on page 128 and “Deleting A User’s SPX Connection to Btrieve” on page 131.

- Resource Usage

  Shows current, peak, and maximum usage statistics for the Btrieve NLM. For more information, see “Monitoring Resources with the Resource Usage Option” on page 132.

- Communication Statistics

  Displays Sequenced Packet Exchange (SPX) protocol statistics for the communications module you have loaded (BSPXCOM, BSPXSTUB, or RSPXSTUB). For more information, see “Monitoring SPX Activity with the Communication Statistics Option” on page 135.

Note When you are using the NetWare Btrieve Monitor utility, the statistics on the File Information, User Information, Resource Usage, and Communication Statistics screens are updated regularly. On the Active Btrieve Files and Active Btrieve Users screens, however, you must press the Insert key to see updated statistics.

While running the Btrieve Monitor utility, you can return to the previous screen at any time by pressing the Esc key.

To exit the utility, press Esc at the Available Options menu. When the Exit window appears, select Yes to verify that you want to exit.

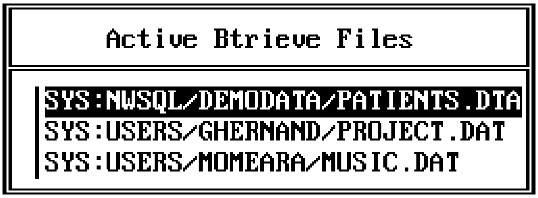
Monitoring Btrieve Files with the Active Resources Option

You can use the Active Resources option on the Available Options menu to perform the following tasks:

- Listing Active (Open) Files
- Displaying Additional Information About an Active File
- Listing All Users Accessing a File

Listing Active (Open) Files

To list all active (that is, open) Btrieve files, select Active Resources from the Available Options menu. The utility displays a scrollable list of active Btrieve files such as the following:



```
Active Btrieve Files

SYS:NWSQL/DEMODATA/PATIENTS.DTA
SYS:USERS/GHERNAND/PROJECT.DAT
SYS:USERS/MOMEARA/MUSIC.DAT
```

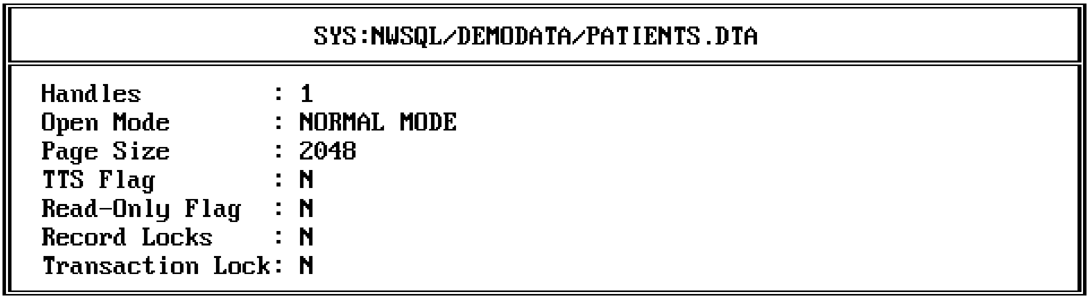
The file pathnames appear in alphabetic order. To update this list, press Insert.

Displaying Additional Information About an Active File

For further information about a particular file, highlight the desired filename listed on the Active Btrieve Files screen and press Enter.

The utility displays a File Information screen similar to the screen shown in Figure 14, providing information about the file you selected. Since the utility dynamically updates the statistics shown on this screen, you may notice the values changing as you view the screen.

Figure 14
Monitor Utility: Active Resources: File Information Screen



```
                    SYS:NWSQL/DEMODATA/PATIENTS.DTA

 Handles         : 1
 Open Mode       : NORMAL MODE
 Page Size       : 2048
 TTS Flag        : N
 Read-Only Flag  : N
 Record Locks    : N
 Transaction Lock: N
```

For an explanation of each field that appears on this screen, refer to Table 2, which describes all the fields associated with the Active Resources option.

Table 2
Active Resources Field Descriptions

Field	Description	
Handles	Shows the number of Btrieve handles the user has as a result of opening files. Btrieve creates a handle each time a user opens a file; therefore, a single user can have several handles for the same file.	
Open Mode	Indicates the mode used to open the file:	
	Accelerated	The application that opened the file has shared, read/write access. With Btrieve 6.*x*, Accelerated mode is equivalent to Normal mode unless the file is flagged transactional.
	Exclusive	The application that opened the file has exclusive access. Other applications cannot open the file until the calling application closes it.
	Normal	The application that opened the file has normal shared, read/write access.
	Read only	The application that opened the file has read-only access; the application cannot modify the file.
Page Size	Indicates the page size (in bytes) of this file. (A page is the smallest unit of storage that Btrieve moves between main memory and the disk.)	
TTS Flag	Indicates whether TTS™ is being used with the file. (For more information on TTS, refer to your NetWare documentation.)	
Read-Only Flag	Indicates whether the file is flagged through NetWare as read only.	

Table 2 *continued*

Active Resources Field Descriptions

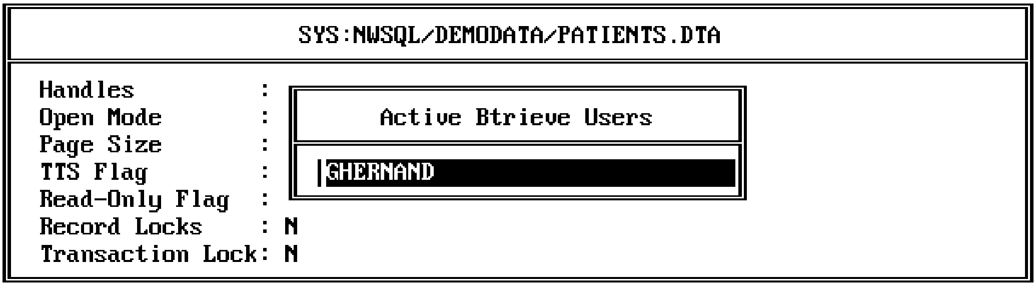
Field	Description
Record Locks	Shows the lock status of the current record: s — Single-record lock outside a transaction S — Single-record lock within a transaction m — Multiple-record lock outside a transaction M — Multiple-record lock within a transaction N — No record locks Single-record locks allow a user to lock only one record at a time. Multiple-record locks allow a user to lock more than one record at a time.
Transaction Lock	Indicates whether the entire file is locked by a transaction. A transactional file lock exists only as long as the application that opened the file is processing a transaction. Y indicates the entire file is locked. N indicates the file is not locked.

Listing All Users Accessing a File

From the Active Resources option's File Information screen, you can display a list of all users currently accessing the designated file.

Press Enter to display a scrollable list of active Btrieve users, which will look similar to the screen shown in Figure 15. (Again, refer to Table 2 for a description of each field on this screen.)

Figure 15
Monitor Utility: Available Options: List of Active Btrieve Users



```
                    SYS:NWSQL/DEMODATA/PATIENTS.DTA

 Handles         :
 Open Mode       :        Active Btrieve Users
 Page Size       :
 TTS Flag        :      GHERNAND
 Read-Only Flag  :
 Record Locks    : N
 Transaction Lock: N
```

Monitoring Btrieve Users with the User Information Option

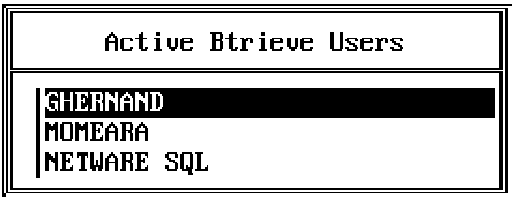
You can use the User Information option on the Available Options menu to perform the following tasks:

- Listing All Active Btrieve Users
- Listing Files Accessed by a Specific User
- Deleting A User's SPX Connection to Btrieve

Listing All Active Btrieve Users

To list all Btrieve users active on the current server, select User Information from the Available Options menu.

The Btrieve Monitor utility displays a scrollable list of active Btrieve users similar to the following:



Active Btrieve Users

GHERNAND
MOMEARA
NETWARE SQL

The utility identifies each active user based on user location, as follows:

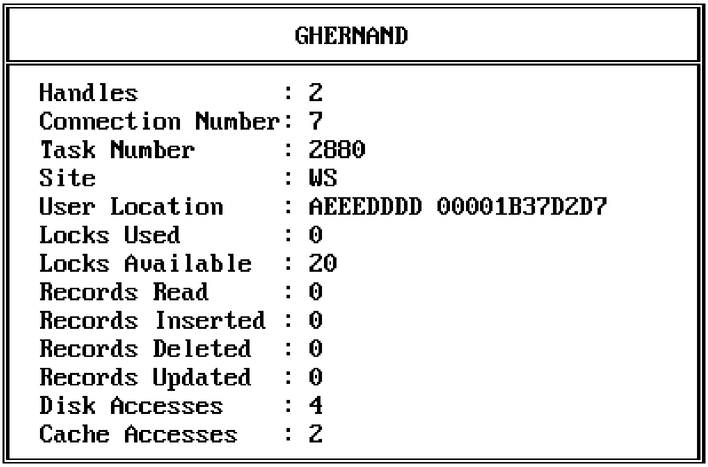
- If the user is located at a workstation, the utility displays the username (such as JDOE).
- If the user is located at a local server, the utility displays either the process-supplied, two-character agent ID or the full name of the process (such as Scalable SQL).
- If the user is located at a remote server, the utility displays either the process-supplied, two-character agent ID or the full name of the process (such as Scalable SQL), provided the utility can determine the full name.

To update the list of active users, press Insert.

To display information about a user, highlight the username on the Active Btrieve Users screen and press Enter. A User Information screen similar to that shown in Figure 16 appears.

Figure 16

Monitor Utility: Available Options: User Information Screen



```
                    GHERNAND

 Handles          : 2
 Connection Number: 7
 Task Number      : 2880
 Site             : WS
 User Location    : AEEEDDDD 00001B37D2D7
 Locks Used       : 0
 Locks Available  : 20
 Records Read     : 0
 Records Inserted : 0
 Records Deleted  : 0
 Records Updated  : 0
 Disk Accesses    : 4
 Cache Accesses   : 2
```

Since the utility dynamically updates the statistics displayed on this screen, you may notice the values changing as you view the information.

Table 3 includes a description of each field that appears on the User Information screen.

Table 3

User Information Field Descriptions

Field	Description
Handles	Shows the number of Btrieve handles the user has as a result of opening files. Btrieve creates a handle each time a user opens a file. A single user can therefore have several handles for the same file.
Connection Number	Displays the NetWare connection number of the process if the process originates at a workstation. If the process originates at a server, this column is empty.
Task Number	Contains the process-supplied task number if the process originates at the server, a MS Windows workstation, or an OS/2 workstation. If the process originates at a DOS workstation, this column contains the SPX socket number.

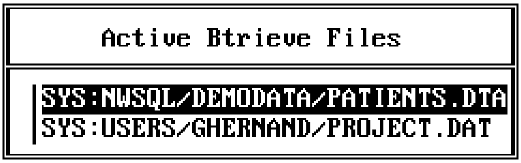
Table 3 *continued*

User Information Field Descriptions

Field	Description
Site	Specifies the location of the user process, as follows: LS —Local server RS —Remote server WS—Workstation
User Location	Identifies the user process, as follows: ◆ If the user is located at a workstation, this column contains the network number and node address. ◆ If the user is located at the server, this column contains the server name.
Locks Used	Indicates the number of locks that the user has explicitly requested. The number of locks in use varies, depending on whether the user is in a transaction, as follows: ◆ If the user is outside a transaction, this number indicates the number of records that the user has explicitly locked. ◆ If the user is inside a concurrent transaction, this number indicates the number of pages in the file that the user has explicitly locked. Although the user actually requests record locks, these are converted to page locks inside a concurrent transaction. Consequently, five record locks are shown as two page locks if the five records are stored on two pages. ◆ If the user is inside an exclusive transaction or the user holds no locks, this number is zero.
Locks Available	Indicates the total number of read locks available to the user.
Records Read	Number of records the user has read.
Records Inserted	Number of records the user has inserted.
Records Deleted	Number of records the user has deleted.
Records Updated	Number of records the user has updated.
Disk Accesses	Shows the number of times the user has made Btrieve calls that required disk I/O.
Cache Accesses	Shows the number of times the user has made Btrieve calls that accessed the Btrieve cache buffers.

Listing Files Accessed by a Specific User

While the User Information screen is displayed, you can press Enter to list all Btrieve files currently accessed by that user. The Active Btrieve Files screen, similar to the following, appears:



```
Active Btrieve Files

SYS:NWSQL/DEMODATA/PATIENTS.DTA
SYS:USERS/GHERNAND/PROJECT.DAT
```

To update the list of active Btrieve files, press Insert.

Deleting A User’s SPX Connection to Btrieve

Deleting a user’s connection removes that user from the list of active Btrieve users and terminates the user’s SPX connection to Btrieve.

Follow these steps to delete a user’s SPX connection:

Procedure

1. **Select the User Information option on the Available Options menu.**
2. **On the Active Btrieve Users screen, highlight the user connection that you want to delete, and press the Delete key.**
3. **In the prompt box that appears, select Yes if you are sure you want to delete the specified user. Otherwise, select No or press Esc.**

Note

If the Btrieve Monitor utility does not list a connection number for the user you want to delete, but the NetWare connection has been terminated, then a Btrieve session is still active for that user. You must restart the workstation that originated the Btrieve session to delete the user for that session.

You can avoid this problem by ensuring that your application issues a Btrieve Reset operation to close active Btrieve files and release all resources held by the application.

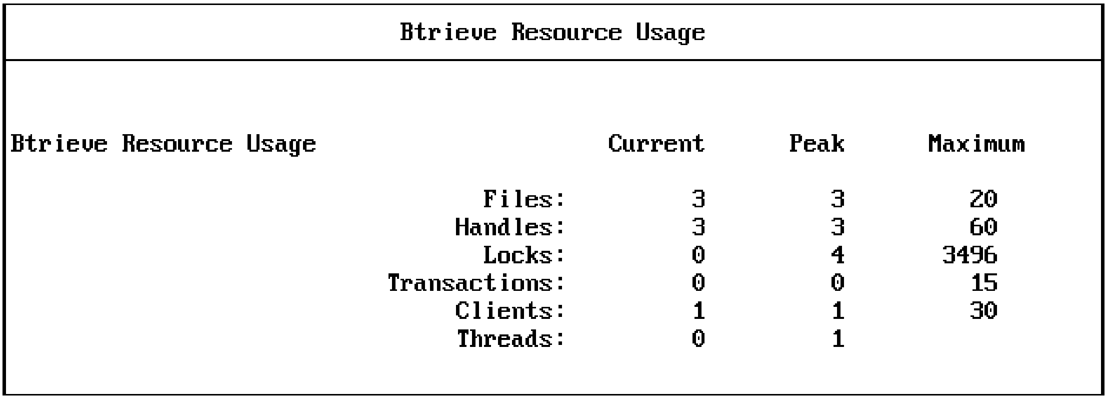
Monitoring Resources with the Resource Usage Option

The Resource Usage option on the Available Options menu shows you (in real time) the total resources in use by the Btrieve NLM since it was loaded.

When you select this option, the Btrieve Monitor utility displays the Btrieve Resource Usage screen, shown in Figure 17.

Figure 17

Monitor Utility: Available Options: Btrieve Resource Usage Screen



```
                         Btrieve Resource Usage

Btrieve Resource Usage                         Current      Peak      Maximum

                                Files:               3         3           20
                              Handles:               3         3           60
                                Locks:               0         4         3496
                         Transactions:               0         0           15
                              Clients:               1         1           30
                              Threads:               0         1
```

Since the utility dynamically updates the statistics shown on this screen, you may notice the totals changing as you view the information.

Note

The Current values on the Btrieve Resource Usage screen are cumulative from the time you first access the screen.

Table 4 includes a description of each field and an explanation of each statistic that appears on the Resource Usage screen.

Table 4
Resource Usage Field Descriptions

Field	Description	Statistics	Explanation
Files	Number of active Btrieve files.	Current	Number of active Btrieve files.
		Peak	Highest number of Btrieve files that have been open simultaneously since the Btrieve NLM was loaded.
		Maximum	Value set for the Number of Open Files configuration option.
Handles	Number of handles issued for Btrieve files.	Current	Number of active file handles.
		Peak	Highest number of handles used simultaneously since the Btrieve NLM was loaded.
		Maximum	Value set for the Number of Handles configuration option.
Locks	Number of implicit (system) page locks involved in concurrent transactions.	Current	Number of implicit active page locks.
		Peak	Highest number of implicit page locks used simultaneously since the Btrieve NLM was loaded.
		Maximum	Maximum simultaneous page locks that the system will allow.
Transactions	Number of concurrent and exclusive transactions.	Current	Number of active concurrent and exclusive transactions.
		Peak	Highest number of transactions active simultaneously since the Btrieve NLM was loaded.
		Maximum	Value set for the Number of Transactions configuration option.

Table 4 *continued*

Resource Usage Field Descriptions

Field	Description	Statistics	Explanation
Clients	Number of Btrieve processes.	Current	Number of active Btrieve processes.
	A process can be a BSPXCOM thread representing a client, a Message Router thread representing a client, or a client NLM on the present server.	Peak	Highest number of processes simultaneously active since the Btrieve NLM was loaded.
		Maximum	Value set for the Number of Remote Sessions configuration option.
Threads	Number of programs or program threads calling Btrieve simultaneously.	Current	Last system snapshot of active programs or program threads calling Btrieve simultaneously.
		Peak	Maximum value ever encountered for the Current value since the NLM was loaded.

Monitoring SPX Activity with the Communication Statistics Option

The Communication Statistics option on the Available Options menu shows you (in real time) the network requests, packet buffers, and sessions in use for the communications module you have loaded.

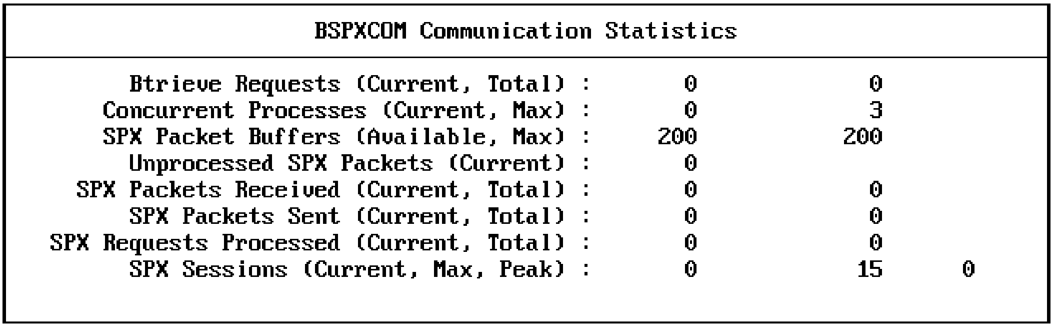
When you select this option, the Btrieve Monitor utility displays the Communications Statistics screen for BSPXCOM, BSPXSTUB, or RSPXSTUB (depending on which communications module is loaded). This screen is similar to that shown in Figure 18.

The communication activity shown on this screen reflects the communication activity of the communications module loaded at your server:

- ◆ If you loaded BSPXCOM, you see incoming and outgoing SPX statistics.
- ◆ If you loaded BSPXSTUB, you see all zeros for the communication statistics.

Figure 18

Monitor Utility: Available Options: Communications Statistics Screen



```
                    BSPXCOM Communication Statistics

         Btrieve Requests (Current, Total) :       0             0
       Concurrent Processes (Current, Max) :       0             3
       SPX Packet Buffers (Available, Max) :     200           200
         Unprocessed SPX Packets (Current) :       0
     SPX Packets Received (Current, Total) :       0             0
         SPX Packets Sent (Current, Total) :       0             0
   SPX Requests Processed (Current, Total) :       0             0
         SPX Sessions (Current, Max, Peak) :       0            15       0
```

- ◆ If you loaded RSPXSTUB, you see incoming and outgoing SPX communication statistics from the Message Router.

This screen does not show the communication activity of any remote NLMs.

Note: The Total values shown on this screen are cumulative from the time Btrieve is loaded. The Current values are cumulative from the time you display the screen.

Table 5 includes a description of each field and an explanation of each statistic that appears on the Communication Statistics screen.

Table 5
Communication Statistics Field Descriptions

Field	Description	Statistics	Explanation
Btrieve Requests	Number of network requests the NLM has processed from workstations or remote server-based applications.	Current	Current value since the last screen update.
		Total	The number of requests received since the Btrieve NLM was loaded.
Concurrent Processes	Number of remote requests the NLM processes at one time.	Current	Current value since the last screen update.
		Max	The maximum number of remote clients the Btrieve NLM can process at one time.
SPX Packet Buffers	Number of SPX packet buffers available to the NLM.	Available	Current number of available packet buffers (as of the last screen update).
		Max	Value set for the maximum number of available packet buffers. Each session is allocated two packet buffers for Btrieve requests.
Unprocessed SPX Packets	Number of SPX packets the NLM has received but not yet processed.	Current	Current value since the last screen update.
SPX Packets Received	Number of SPX packets the NLM has received from applications.	Current	Current value since the last screen update.
		Total	The number of packets received since the communications NLM was loaded.

Table 5 *continued*

Communication Statistics Field Descriptions

Field	Description	Statistics	Explanation
SPX Packets Sent	Number of SPX packets the NLM has sent to other applications.	Current	Current value since the last screen update.
		Total	The number of packets sent since the communications NLM was loaded. The total packets sent may be larger than the total packets received because a single request received might produce several packets sent.
SPX Requests Processed	Number of SPX requests the NLM has processed.	Current	Current value since the last screen update.
		Total	The number of requests processed since the communications NLM was loaded.
SPX Sessions	Number of remote clients who have established SPX sessions with the communications NLM.	Current	Current value since the communications NLM was loaded.
		Max	Value set for the Number of Remote Sessions configuration option.
		Peak	Highest value since the communications NLM was loaded. This value indicates the highest number of SPX clients that have simultaneously had concurrent active sessions with the communications NLM.

NetWare Btrieve Maintenance Utility

The Btrieve Maintenance utility (BUTIL.NLM) is a command-line utility that allows you to create, manipulate, and recover Btrieve data files using the BUTIL commands.

The Maintenance utility runs as an NLM at the server. You can access it at the server console or at a workstation through the RCONSOLE utility. You can execute the Maintenance utility commands from the server command line or from a command file.

This section discusses the following topics:

- "System Requirements for the NetWare Btrieve Maintenance Utility" on page 138
- "Maintenance Utility Overview" on page 139
- "NetWare Btrieve Maintenance Utility Commands" on page 143

System Requirements for the NetWare Btrieve Maintenance Utility

To run the Maintenance utility, be sure that the following are installed on your server:

Checklist

❑ NetWare® 4 or later

Running NetWare Btrieve 6.1*x* in a NetWare 3.11 or 3.12 environment requires the AFTER311.NLM, which Btrieve loads automatically.

❑ Btrieve 6.1*x* or later

In addition, make sure the following files are loaded in the SYS:SYSTEM directory of the server (the NetWare INSTALL utility places them there automatically during installation):

❑ BUTIL.NLM—NetWare Btrieve 6.1*x* Maintenance utility

❑ BUTIL.MSG—Message file for the Maintenance utility

Maintenance Utility Overview

This section provides information you need to know before using the NetWare Btrieve Maintenance utility commands. It discusses the following topics:

- "Command Format" on page 139
- "Fundamental Concepts for Using the Maintenance Utility Commands" on page 140
- "Command Files" on page 142

Note: If you have used the utility before, you may want to skip to the individual command discussions in "NetWare Btrieve Maintenance Utility Commands" on page 143. The commands are discussed in alphabetic order.

Command Format

The format for the Maintenance utility commands is as follows:

```
LOAD BUTIL [–command [parameter ...]] |
  [@commandFile]
```

–command

A Maintenance utility command, such as COPY.

You must precede the command with a space followed by a dash (–).

Table 6 lists the Maintenance utility commands.

parameter

Information that the command may require.

The discussions of the individual commands (in NetWare Btrieve Maintenance Utility Commands on page 143) provide details when applicable.

@commandFile

Full pathname of a NetWare Command file.

The commands are listed and their functions are summarized in Table 6 on page 143.

Fundamental Concepts for Using the Maintenance Utility Commands

The following paragraphs describe concepts you should understand before using the Maintenance utility commands. For more detailed information about these concepts, see Appendix A, “Btrieve Concepts,” and Appendix B, “Description Files.”

Filenames

The Maintenance utility runs as an NLM. The NLM environment does not recognize drive letters or environment variables. Thus, for commands that require a filename, the name must include the full pathname, such as SYS:\NWSQL\DEMODATA\PATIENTS.DTA. If you do not specify a volume, the utility assumes SYS: is the volume.

Owner Names

Btrieve allows you to restrict access to a file by specifying an owner name. Since owner names are optional, the files you use with this utility may or may not require an owner name.

If the file requires an owner name, you must specify it using the /O option. (You can also use a dash with this option, as in –O.) Owner names *are* case sensitive; that is, Sandy and SANDY are not considered to be the same. You can follow the /O option with an owner name or an asterisk (*). If you use an asterisk, the utility prompts you for an owner name.

Description Files

A *description file* is an ASCII file containing information that the Maintenance utility commands CREATE, INDEX, and SINDEX need to perform their operations.

Description files contain one or more *elements*. An element consists of a keyword, followed by an equal sign (=), followed by a value (with no space separator). Each element in the description file corresponds to a particular characteristic of a Btrieve file or key definition. For additional information, see “Sample Description File for the CREATE Command” on page 150.

Continuous Operation

Continuous operation is a Btrieve feature that allows you to back up files while they are in use by Btrieve applications. Two Maintenance utility commands, STARTBU and ENDBU, begin and end continuous operation on a file or set of files.

When continuous operation begins, Btrieve creates a temporary Btrieve file (called a *delta file*) for each data file in order to record any changes made to that file while the backup is taking place. This temporary file may surpass the size of the original data file if users make extensive changes to the file during continuous operation.

When continuous operation ends, Btrieve updates the master Btrieve files with changes stored in the delta files. Btrieve deletes the delta files as soon as all applications close the corresponding Btrieve files.

Note

As indicated above, when you place a Btrieve file into continuous operation mode, Btrieve creates a temporary file with the same name as the data file but with a .^^^ extension. Therefore, do not create multiple Btrieve files with the same names but different extensions. For example, do not use a naming scheme such as INVOICE.HDR and INVOICE.DET for your Btrieve files.

To back up files using continuous operation, issue the BUTIL -STARTBU command followed by the filename or set of files. Then run your backup program. To stop continuous operation, issue the BUTIL -ENDBY command. (See “ENDBU” on page 153 for additional information about this operation.)

Hint

The best time to place Btrieve data files into continuous operation for backup is when the fewest users will be making modifications to the files.

Command Files

You can use a command file to perform either of the following tasks:

◆ Execute a command that is too long to fit on the command line.

◆ Execute a command that you use often by entering the command once in the command file and then executing the command file as often as you want.

Command files contain the same information as that required on the command line.

Rules for Command Files

Observe the following rules when creating a Maintenance utility command file:

Checklist

❑ You must limit each line to 130 characters.

Lines longer than 130 characters could cause the server to end abnormally. For this reason, do not place long Maintenance utility commands in a server command (.NCF) file.

❑ You cannot split a single parameter across two lines.

❑ You can specify only one command per command file.

Command File Example

The following is an example command file, COPYPATS.CMD. The file calls the BUTIL –CLONE command to clone the file PATIENTS.DTA.

```
–clone sys:\nwsql\demodata\allpats.dta
sys:\nwsql\demodata\patients.dta
```

The following command uses the COPYPATS.CMD file:

```
load butil @sys:\nwsql\demodata\copypats.cmd
```

NetWare Btrieve Maintenance Utility Commands

This section describes the Maintenance utility commands and explains when and how to use each one. Table 6 lists each command in alphabetical order and briefly describes what each command does. Following that is a discussion of each individual command.

Table 6
Maintenance Utility Commands

Command	Description
CLONE	Creates a new, empty Btrieve file using an existing file’s specifications and key definitions.
CLROWNER	Clears (drops) the owner name associated with a Btrieve file.
COPY	Copies the contents of one Btrieve file to another.
CREATE	Creates a Btrieve file.
DROP	Drops an index.
ENDBU	Ends continuous operation on Btrieve files defined for backup.
INDEX	Creates an external index file.
LOAD	Loads the contents of a sequential file (ASCII) into a Btrieve file.
RECOVER	Reads data sequentially from a Btrieve file and writes the results to a sequential (ASCII) file.
SALVAGE	Examines a file’s Page Allocation Table (PAT) to determine if corruption has occurred and if repair is required.
SAVE	Reads data along a key path and writes the results to a sequential (ASCII) file.
SETOWNER	Assigns an owner name to a Btrieve file.
SINDEX	Creates an index.
STARTBU	Starts continuous operation on files defined for backup.
STAT	Reports statistics about file attributes and current sizes of Btrieve files.
VER	Displays the version of the Btrieve NLM that is loaded at the server.

At any time while using the Maintenance utility commands, you can enter the following to display a list of these commands:

```
load butil
```

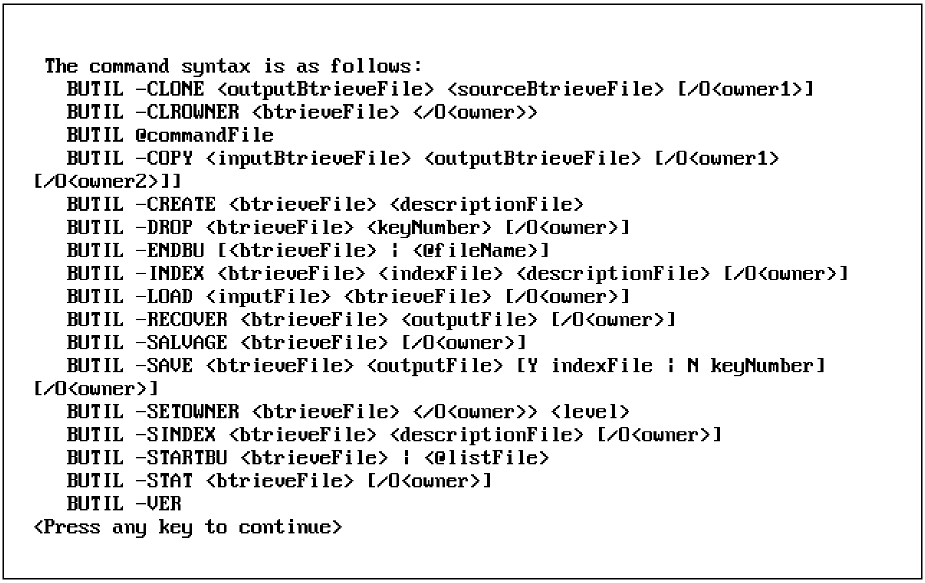
The utility responds with the screen shown in Figure 19.

CLONE

The CLONE command creates a new, empty Btrieve file with the same file specifications as an existing file including any supplemental indexes, but excluding the owner name.

The new Btrieve file includes all the defined key characteristics, such as key position, key length, or duplicate key values, contained in the existing, original file.

Figure 19
Maintenance Utility: Command Syntax Screen



```
 The command syntax is as follows:
   BUTIL -CLONE <outputBtrieveFile> <sourceBtrieveFile> [/O<owner1>]
   BUTIL -CLROWNER <btrieveFile> </O<owner>>
   BUTIL @commandFile
   BUTIL -COPY <inputBtrieveFile> <outputBtrieveFile> [/O<owner1>
[/O<owner2>]]
   BUTIL -CREATE <btrieveFile> <descriptionFile>
   BUTIL -DROP <btrieveFile> <keyNumber> [/O<owner>]
   BUTIL -ENDBU [<btrieveFile> | <@fileName>]
   BUTIL -INDEX <btrieveFile> <indexFile> <descriptionFile> [/O<owner>]
   BUTIL -LOAD <inputFile> <btrieveFile> [/O<owner>]
   BUTIL -RECOVER <btrieveFile> <outputFile> [/O<owner>]
   BUTIL -SALVAGE <btrieveFile> [/O<owner>]
   BUTIL -SAVE <btrieveFile> <outputFile> [Y indexFile | N keyNumber]
[/O<owner>]
   BUTIL -SETOWNER <btrieveFile> </O<owner>> <level>
   BUTIL -SINDEX <btrieveFile> <descriptionFile> [/O<owner>]
   BUTIL -STARTBU <btrieveFile> | <@listFile>
   BUTIL -STAT <btrieveFile> [/O<owner>]
   BUTIL -VER
<Press any key to continue>
```

Format

```
LOAD BUTIL –CLONE outputBtrvFile sourceBtrvFile
  [/Oowner]
```

outputBtrvFile The full pathname you want to use for the new, empty Btrieve file.

sourceBtrvFile The full pathname of the existing, original Btrieve file you want to replicate.

owner The owner name, if any, for the source Btrieve file. The new Btrieve file will not have an owner name.

Remarks

Btrieve 6.*x* allows a maximum of 23 key segments in a Btrieve file with a page size of 1,024 bytes. Therefore, the CLONE command sets the page size in the new Btrieve file to 2,048 bytes if the existing Btrieve file contains 24 key segments and has a page size of 1,024 bytes.

This expansion occurs if the existing Btrieve file has a format earlier than Btrieve 6.0 and the Btrieve NLM was not loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option.

Example

The following command creates the NEWAPP.DTA file by cloning the PATIENTS.DTA file.

```
load built -clone sys:\nwsql\demodata\newapp.dta
sys:\nwsql\demodata\patients.dta
```

Hint

If you are trying to recover from receiving a Status Code 30 (The specified file is not a Btrieve file) and you suspect that the header page of the source file might be damaged, try creating the new Btrieve file by using the BUTIL – CREATE command with a description file.

CLROWNER

The CLROWNER command clears (drops) the owner name of a Btrieve file.

Format

LOAD BUTIL –CLROWNER *btrvFile* /**O***owner*

btrvFile	The full pathname of the Btrieve file.
owner	The owner name to be cleared (dropped).

Example

The following command clears the owner name for the PATIENT1.DTA file. The owner name for the file is Sandy.

```
load butil -clrowner
sys:\nwsql\demodata\patient1.dta /OSandy
```

COPY

The COPY command copies the contents of one Btrieve file to another.

COPY retrieves each record in the input Btrieve file and inserts it into the output Btrieve file. The record size must be the same in both files. After copying the records, COPY displays the total number of records inserted into the new Btrieve file.

Note: COPY performs in a single step the same functions as a RECOVER command followed by a LOAD command.

Format

```
LOAD BUTIL –COPY inputBtrvFile outputBtrvFile
  [/Oowner1 [/Oowner2]]
```

inputBtrvFile — The full pathname of the Btrieve file from which you want to transfer records.

outputBtrvFile — The full pathname of the Btrieve file into which you want to insert records. The output Btrieve file can contain data or be empty.

owner1 — The owner name of the input Btrieve file, if required.

If only the output Btrieve file requires an owner name, specify /O followed by a blank for `owner1` (as illustrated in the example).

owner2 — The owner name of the output Btrieve file, if required.

Remarks

Using the COPY command, you can create a Btrieve file that contains data from an old file but has new key characteristics.

To do so, proceed as follows:

Procedure

1. **Use the CREATE command to create an empty Btrieve file with the desired key characteristics (key position, key length, or duplicate key values).**

2. **Use the COPY command to copy the contents of the existing Btrieve file into the newly created Btrieve file.**

Example

The following command copies the records in PATIENTS.DTA to NEWPATS.DTA. The PATIENTS.DTA input file does not require an owner name, but the NEWPATS.DTA output file uses the owner name Pam.

```
load butil -copy
sys:\nwsql\demodata\patients.dta
sys:\nwsql\demodata\newpats.dta /O /OPam
```

If you omit the first /O from this example, the utility assumes that the owner name Pam belongs to the input Btrieve file, not the output Btrieve file.

CREATE

The CREATE command generates an empty Btrieve file using the characteristics you specify in a description file.

Before you can use the CREATE command, you must create a description file to specify the new key characteristics. For more information, see Appendix B, “Description Files.”

Format

LOAD BUTIL –CREATE *btrvFile descriptionFile*

btrvFile	The full pathname of the Btrieve file you want to create. If the pathname is the name of an existing Btrieve file, this command creates a new, empty Btrieve file in place of the existing Btrieve file. Any data that was stored in the existing Btrieve file is lost and cannot be recovered. However, Btrieve does not create a new Btrieve file if you specify `replace=n` in the description file. (For an example, see “Replace Existing File” in Appendix B.)
descriptionFile	The full pathname of the description file containing the specifications for the new Btrieve file.

Example

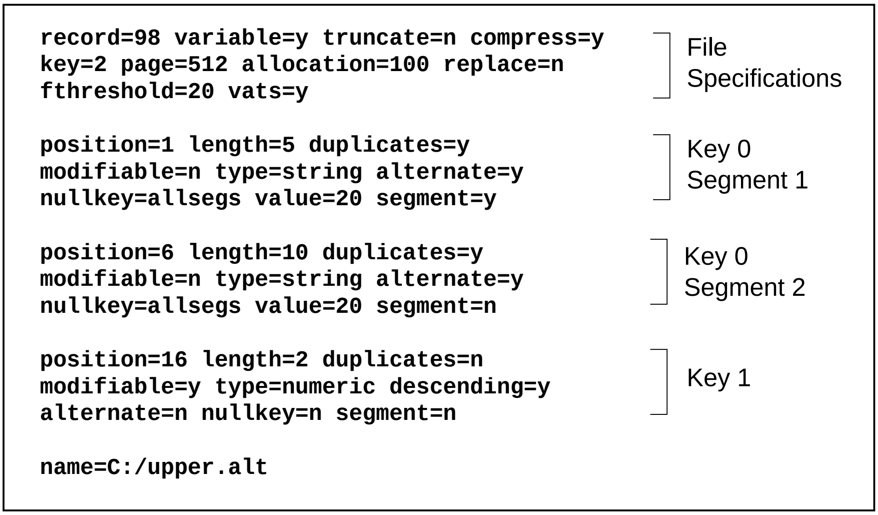
The following command creates a Btrieve file named PATIENTS.DTA using the description provided in the BUILD.DES description file.

```
load butil -create sys:\nwsql\patients.dta
sys:\nwsql\build.des
```

Sample Description File for the CREATE Command

The sample description file shown in Figure 20 creates a Btrieve file with a page size of 512 bytes (page =512). The fixed-length portion of the record is 98 bytes long (record=98).

Figure 20
Sample Description File for the BUTIL -CREATE Command



```
record=98 variable=y truncate=n compress=y
key=2 page=512 allocation=100 replace=n
fthreshold=20 vats=y

position=1 length=5 duplicates=y
modifiable=n type=string alternate=y
nullkey=allsegs value=20 segment=y

position=6 length=10 duplicates=y
modifiable=n type=string alternate=y
nullkey=allsegs value=20 segment=n

position=16 length=2 duplicates=n
modifiable=y type=numeric descending=y
alternate=n nullkey=n segment=n

name=C:/upper.alt
```

The file also specifies the following:

- Variable-length records (variable=y)
- No blank truncation (truncate=n)
- Data compression (compress=y)
- Free space threshold of 20 percent (fthreshold=20)
- Variable-tail Allocation Tables (vats=y)

Btrieve preallocates 100 pages, or 51,200 bytes, when it creates the file (allocate=100). The file has two keys: Key 0 and Key 1 (key=2).

Key 0 is a segmented key with two segments. The first segment of Key 0 has the following key *attributes*, or characteristics:

- Begins at the first byte in the record (position=1)
- Is 5 bytes long (length=5)
- Allows more than one record to have the same value for that key (duplicates=y)
- Does not allow the key to be modified (modifiable=n)
- Defines the key type as an alphanumeric string (type=string)
- Specifies that the key uses an alternate collating sequence (alternate=y) that is defined in the ACS file UPPER.ALT (name=*path*/upper.alt)
- Specifies excluding from the index those records in which *all* the bytes in *all segments* of a key contain a null value (nullkey=allsegs)
- Defines the null value as an ASCII blank character, or space (value=20)
- Specifies a second segment (segment=y)

The second segment of Key 0 has the same key attributes as the first segment, except that it begins at the sixth byte in the record (position=6), is 10 bytes long (length=10), and is not followed by another segment (segment=no).

Key 1, which begins at the sixteenth byte and is two bytes long, is a numeric, nonsegmented key that does not allow duplicates but permits modification. It is sorted in descending order.

For detailed information about file and key attributes, refer to Appendix B, “Description Files.”

DROP

The DROP command removes an index from a Btrieve file and adjusts the key numbers of any remaining indexes, subtracting 1 from each subsequent key number.

If you do not want to renumber the keys, you can add a *bias* of 128 to the key number you specify to be dropped. This renumbering feature is available only for Btrieve 6.*x* files.

Format

LOAD BUTIL –DROP *btrvFile keyNumber* [/**O***owner*]

btrvFile	The full pathname of the Btrieve file from which you are dropping the index.
keyNumber	The number of the index you want to remove. If you want to preserve the original key numbers, add a 128 bias to the key number you specify.
owner	The owner name for the Btrieve file, if required.

Examples

In both of the following examples, PATIENTS.DTA has three keys. The original keys in the file were numbered 0, 1, and 2.

In the first example, the BUTIL –DROP command drops key number 1 from the PATIENTS.DTA file and renumbers the remaining keys as 0 and 1.

```
load butil -drop sys:\nwsql\demodata\patients.dta 1
```

In the following example, the DROP command drops key number 1 and does not renumber the keys (indicated by the number 129—key number 1 plus the 128 bias). The key numbers remain 0 and 2.

```
load butil -drop
sys:\nwsql\demodata\patients.dta 129
```

ENDBU

The ENDBU command ends *continuous operation* on a Btrieve file or set of Btrieve files previously defined for backup.

Execute this command after you have issued the STARTBU command and your backup utility has finished running. (For more information on the STARTBU command, see “STARTBU” on page 167. For more information on continuous operation, see “Continuous Operation” on page 141.)

Procedure

1. **To back up Btrieve files using continuous operation, first issue the command, followed by the Btrieve filename or the name of the text file containing the list of file you want to place in continuous operation:**

   ```
   load butil -startbu <btrvFile | @filename>]
   ```

2. **Run your backup program.**

3. **Stop continuous operation by issuing the following command:**

   ```
   load butil -endbu [btrvFile | @filename]
   ```

Note

When you place a Btrieve file into continuous operation mode, Btrieve creates a temporary file with the same name as the Btrieve data file, but with a .^^^ extension. Therefore, do not create multiple Btrieve files with the same names but different extensions. For example, do not use a naming scheme such as INVOICE.HDR and INVOICE.DET for your Btrieve files.

Format

```
LOAD BUTIL –ENDBU [btrvFile | @filename]
```

btrvFile The full pathname of the Btrieve file for which you want to end continuous operation.

@filename The name of a text file containing a list of Btrieve files for which you want to end continuous operation.

The text file must contain the full pathname for each Btrieve file, separated by a space or carriage return/ line feed. This list of files is typically the same as the list used with the STARTBU command.

If you do not specify any Btrieve files when issuing the LOAD BUTIL –ENDBU command, the utility stops continuous operation on all Btrieve files initialized by BUTIL –STARTBU and currently running in continuous operation mode.

Example

The following example ends continuous operation on the PATIENTS.DTA file.

```
load butil -endbu sys:\nwsql\demodata\patients.dta
```

INDEX

The INDEX command builds an external index file for an existing Btrieve file, based on a field not previously specified as a key in the existing file.

Before you can use the INDEX command, you must create a description file to specify the new key characteristics. (For more information on description files, see Appendix B, "Description Files.")

The external index file created is a *key-only* Btrieve file. The records in the new file consist of the following:

- The 4-byte address of each record in the existing Btrieve file
- The new key value on which you want to sort

Note: If the key length you specify in the description file is 10 bytes, the record length of the external index file would be 14 bytes (10 plus the 4-byte address).

Format

```
LOAD BUTIL –INDEX btrvFile indexFile descriptionFile
  [/Oowner]
```

btrvFile The full pathname of the existing Btrieve file for which you want to build an external index.

indexFile The full pathname of the index file in which Btrieve should store the external index.

descriptionFile The full pathname of the description file you have created containing the new key definition.

The description file should contain a definition for each segment of the new key.

owner The owner name for the Btrieve file, if required.

Remarks

The INDEX command creates the external index file and then displays the number of records that were indexed. If you want to retrieve the Btrieve file's records using the external index file, use the SAVE command (described in "SAVE" on page 163).

Sample Description File for the INDEX Command

The description file shown in Figure 21 illustration defines a new key with one segment.

The key begins at byte 30 of the record and is 10 bytes long. It allows duplicates, is modifiable, is a `string` type, and uses no alternate collating sequence.

Figure 21
Sample Description File for the INDEX Command

```
position=30 length=10 duplicates=y modifiable=y
type=string alternate=n segment=n
```

Example

The following command creates an external index file called NEWPAT.IDX using a Btrieve file called PATIENTS.DTA. The PATIENTS.DTA file does not require an owner name.

The description file containing the definition for the new key is called NEWIDX.DES.

```
load butil -index
sys:\nwsql\demodata\patients.dta
sys:\nwsql\demodata\newpat.idx
sys:\nwsql\demodata\newidx.des
```

LOAD

The LOAD command inserts records from an input sequential (ASCII) file into a Btrieve file.

LOAD performs no conversion on the data in the input sequential file. After the utility transfers the records to the Btrieve file, it displays the total number of records loaded.

Before running the LOAD command, you must create the input sequential file and the Btrieve file. You can create the input sequential file, in the required format (as explained subsequently), using a standard text editor or an application. You can create the Btrieve file using either BUTIL –CREATE or BUTIL –CLONE.

Format

LOAD BUTIL –LOAD *inputFile btrvFile* [/**O***owner*]

inputFile	The full pathname of the ASCII sequential file containing the records to be loaded into a Btrieve file.
btrvFile	The full pathname of the Btrieve file into which you want to insert the records.
owner	The owner name for the Btrieve file, if required.

Required File Format

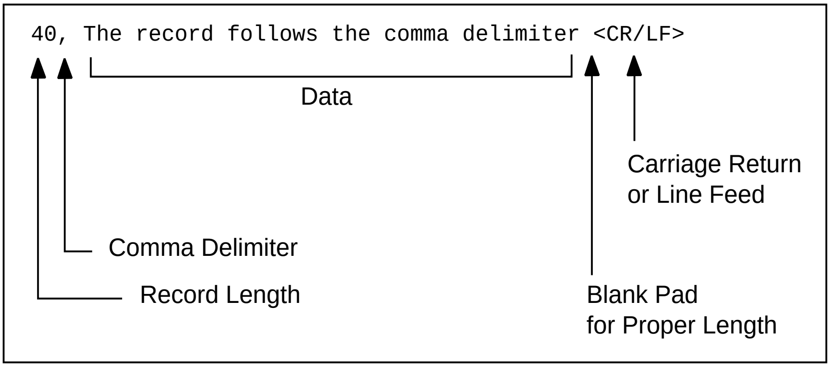
Records in the input sequential file must be in the following format:

- The first field must be a left-adjusted integer (in ASCII) that provides the length of the record. This field does not include the carriage return or line feed.
    - For files with fixed-length records, the length you specify should equal the record length of the Btrieve file.
    - For files with variable-length records, the length you specify must be at least as long as the minimum fixed length of the Btrieve file.
- A separator (either a comma or a blank) must follow the length field.
- The record data follows the separator. The length of the data must be the exact number of bytes specified by the length field.
- A carriage return/line feed (0D0A hexadecimal) must terminate each line. The carriage return/line feed is not included in the length value at the beginning of the line, and LOAD does not insert the carriage return/line feed into the Btrieve file.
- The last line in the file must consist of the end-of-file character (Ctrl+Z or 1A hexadecimal). The SAVE and RECOVER commands and most text editors automatically insert this character in the file.

You can create an input sequential file using either a text editor or an application, as follows:

- If you use a text editor to create the input sequential file, pad each record with blank spaces as necessary to fill the record to the length you specified at the beginning of the record. Fields containing binary data cannot be edited with most text editors.
- If you use an application to create the input sequential file, append a carriage return/line feed to the end of each record and include an end-of-file character (Ctrl+Z or 1A hexadecimal) as the last line in the file. The sequential I/O calls provided by most high-level language processors insert carriage return, line feed, and end-of-file characters automatically.

The following illustration shows the correct format for records in the input sequential file. For this example, the Btrieve file has a defined record length of 40 bytes.



Example

The following example loads sequential records from the PATIENTS.ADR file into the PATIENTS.DTA file. The owner name of the PATIENTS.DTA file is Sandy.

```
load butil -load
sys:\nwsql\demodata\patients.adr
sys:\nwsql\demodata\patients.dta /OSandy
```

RECOVER

The RECOVER command extracts data from a Btrieve file and places it in a sequential (ASCII) file that has the same format as the input sequential file used by the LOAD command.

This command is often useful for extracting some or all of the data from a damaged Btrieve file. The RECOVER command may be able to retrieve many, if not all, the file's records. You can then use the LOAD command to insert the recovered records into a new, undamaged Btrieve file.

Note The Maintenance utility performs no conversion on the data in the records. Therefore, if you use a text editor to modify an output file containing binary data, be aware that some text editors may change the binary data, causing the results to be unpredictable.

The RECOVER command performs the following actions:

- Checks the file's Page Allocation Table (PAT) and reconstructs it, if you request this. The PAT is the part of the Btrieve file that maintains a map of each page's physical location.
- Reads records in physical order from the Btrieve file, using Btrieve Step operations.
- Creates a sequential file that is compatible with the required format for the LOAD command. (See "Required File Format" on page 157 for more information about the format.)
- Displays the total number of recovered records.

Format

LOAD BUTIL –RECOVER *btrvFile outputFile* [/**O***owner*]

btrvFile The full pathname of the Btrieve file from which you want to recover data.

outputFile The full pathname of the ASCII sequential file where you want the utility to store the recovered records.

owner The owner name for the Btrieve file, if required.

Remarks

If the file’s PAT is damaged, a prompt similar to the following appears:

```
The file’s Page Allocation Table seems to be damaged.
BUTIL *strongly* recommends that you make a backup
copy before continuing.
Continue? 1=Yes 2=No
```

By default, the prompt displays 2 (indicating No) on the next line. This allows you to exit the RECOVER command and back up the Btrieve file before proceeding. If you have already backed up the Btrieve file, enter 1 to continue running the RECOVER command.

The RECOVER command allows you to set the Btrieve file’s page size. It displays the following prompt:

```
Enter the page size or 0 to quit: 512
```

The value displayed at this prompt is the result of an attempt to determine the original page size of the Btrieve file. If this value is incorrect, enter the correct page size.

If you enter a page size that differs from the original page size, the result is unpredictable. If you are unsure of the correct page size, change the value as prompted by the utility.

Note

NetWare Btrieve specifies that 6.0*x* and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

If the logical disk drive containing your output sequential file becomes full before the entire Btrieve file has been recovered, the utility stops, indicates the number of records already recovered, and displays the following prompt:

```
The disk volume is full.
Enter new file name to continue or a period to quit.
```

To continue running the RECOVER command using an additional output sequential file, complete one of the following steps:

- If you are recovering the Btrieve file to diskettes, remove the full diskette and replace it with another formatted diskette.
- If you are recovering the Btrieve file to a hard disk, specify another logical disk drive that has space available.

In either case, enter the full pathname of the Btrieve file you want to use to continue storing records, and then press Enter. The utility continues copying records from the Btrieve file to the new output sequential file. This process creates multiple sequential files that you must load separately with the LOAD command.

If the RECOVER command receives a variable page error (Status Code 54), it places all the data it can obtain from the current record in the output sequential file and continues the recovery process.

Upon completion, the utility displays a message similar to the following:

```
16 records recovered.
Operation completed successfully.
```

Example

The following example extracts records from the PATIENTS.DTA file and writes them into the SEQPAT.DAT file.

```
load butil -recover sys:\nwsql\patients.dta
sys:\nwsql\seqpat.dat
```

SALVAGE

The SALVAGE command examines the records in a file’s Page Allocation Table (PAT) to determine whether corruption has occurred. (The PAT maintains a map of the physical location of each page in the Btrieve file.) If corruption has occurred, the utility asks if you want to repair the PAT.

Format

```
LOAD BUTIL –SALVAGE btrvFile [/Oowner]
```

btrvFile The full pathname of the Btrieve file containing the records you want to check.

owner The owner name for the Btrieve file, if required.

Remarks

If the file’s PAT is damaged, the utility reminds you that you should have a backup of the Btrieve file before proceeding and asks if you want to repair the file now. If you have already backed up the Btrieve file, enter Y. If you have not backed up the Btrieve file, enter N.

After you enter Y, the utility asks you to enter a page size and provides you with the result of its attempt to determine the original page size. If you suspect that the value shown is incorrect, enter a new value.

The utility then attempts to repair the Btrieve file, using the new value. If the utility cannot repair the Btrieve file, it sends a message identifying the reason why.

Note The SALVAGE command does not save the records to a sequential file. You must use the LOAD command to perform that operation.

SAVE

The SAVE command retrieves records from a Btrieve file using a specified index path and places them in a sequential file that is compatible with the required format for the LOAD command.

You can then edit the sequential file and use the LOAD command to store the edited data in another Btrieve file. (See “LOAD” on page 156 for more information about the LOAD command.)

SAVE generates a single record in the output sequential file for each record in the input Btrieve file. Upon completion, SAVE displays the total number of records saved.

Note: The Maintenance utility performs no conversion on the data in the records. Therefore, if you use a text editor to modify an output file containing binary data, be aware that some text editors may change the binary data, causing the results to be unpredictable.

Format

```
LOAD BUTIL –SAVE btrvFile outputFile
  [Y indexFile | N keyNumber] [/Oowner]
```

btrvFile The full pathname of the Btrieve file containing the records you want to save.

outputFile The full pathname of the ASCII sequential file in which you want the utility to store the records.

indexFile The full pathname of an external index file by which you want to save records *if* you do not want to save records using the default of the lowest key number.

keyNumber The key number (other than 0) by which you want to save records *if* you do not want to save records using the default of the lowest key number.

owner The owner name for the Btrieve file, if required.

Remarks

If the logical disk drive containing your output sequential file becomes full before the entire Btrieve file has been saved, the utility stops, indicates the number of records already saved, and displays the following message:

```
The disk volume is full.
Enter new file name to continue or a period to quit.
```

To continue the SAVE operation in another output sequential file, complete one of the following steps:

- If you are saving the Btrieve file to diskettes, remove the full diskette and replace it with another formatted diskette.

- If you are saving the Btrieve file to a hard disk, specify another logical disk drive that has space available.

In either case, enter the full pathname of the Btrieve file you want to use to continue storing records, and press Enter. The utility continues copying records from the Btrieve file to the new output sequential file. Keep in mind that this process creates multiple sequential files that you must load separately with the LOAD command.

Examples

The following two examples illustrate how to use the SAVE command to retrieve records from a Btrieve file.

The first example uses the NEWPAT.IDX external index file to retrieve records from the PATIENTS.DTA file and store them in an unformatted text file called PATIENTS.SAV:

```
load butil -save
sys:\nwsql\demodata\patients.dta
sys:\nwsql\demodata\patients.sav
sys:\nwsql\demodata\newpat.idx
```

The next example retrieves records from the PATIENTS.DTA file using key number 3 and stores them in an unformatted text file called PATIENTS.SAV:

```
load butil -save
sys:\nwsql\demodata\patients.dta
sys:\nwsql\demodata\patients.sav N 3
```

SETOWNER

The SETOWNER command creates an owner for a Btrieve file.

Format

`LOAD BUTIL –SETOWNER` *`btrvFile /`***`O`***`owner level`*

btrvFile	The full pathname of the Btrieve file.
owner	The owner name to be set.
level	The type of access restriction for the Btrieve file. The possible values for this parameter are as follows: 0—Requires an owner name for any access mode (no data encryption) 1—Permits read access without an owner name (no data encryption) 2—Requires an owner name for any access mode (with data encryption) 3—Permits read access without an owner name (with data encryption)

Example

The following example creates an owner for the PATIENTS.DTA file. The owner name is Sandy, and the restriction level is 1.

```
load butil -setowner
sys:\nwsql\demodata\patients.dta /OSandy 1
```

SINDEX

The SINDEX command creates an additional index (called a *supplemental index* in previous releases) for an existing Btrieve file.

The key number of the new index is one higher than the previous highest key number for the Btrieve file. An exception to this numbering occurs when a DROP command previously removed an index without renumbering the remaining keys, thus producing an unused key number. In this case, the new index receives the first unused number.

Before you can use the SINDEX command, you must create a description file to define key specifications for the index. For more information on description files, see Appendix B, “Description Files.”

Format

LOAD BUTIL –SINDEX *btrvFile descriptionFile* [/**O***owner*]

btrvFile	The full pathname of the Btrieve file for which you are creating the index.
descriptionFile	The full pathname of the description file containing the description of the index you want to create.
owner	The owner name for the Btrieve file, if required.

Examples

The following example adds an index to the PATIENTS.DTA file. The name of the description file is SUPPIDX.DES.

```
load butil -sindex
sys:\nwsql\demodata\patients.dta
sys:\nwsql\suppidx.des
```

STARTBU

The STARTBU command places a file or set of files into continuous operation for backup purposes.

To back up files using continuous operation, first issue the LOAD BUTIL –STARTBU command, followed by the Btrieve file or set of Btrieve files. Next, run your backup program. Then, issue the LOAD BUTIL –ENDBU command to stop continuous operation.

For additional information on the ENDBU command, see “ENDBU” on page 153. For more information on continuous operation, see “Continuous Operation” on page 141.

Hint

When you place a Btrieve file into continuous operation mode, Btrieve creates a temporary file with the same name as the data file, but with a .^^^ extension. Therefore, do not create multiple Btrieve files with the same names but different extensions. For example, do not use a naming scheme such as INVOICE.HDR and INVOICE.DET for your Btrieve files.

Format

LOAD BUTIL –STARTBU <*btrvFile* | *@filename*>

btrvFile	The full pathname of the Btrieve file on which to begin continuous operation for backup.
@filename	The name of a text file containing the full pathnames of files on which to begin continuous operation. Separate these pathnames with a space or a carriage return/line feed.

Important

This command begins continuous operation only on the files you specify. You cannot use wildcards with the STARTBU command.

Example

The following example starts continuous operation on the PATIENTS.DTA file.

```
load butil -startbu
sys:\nwsql\demodata\patients.dta
```

STAT

The STAT command reports the defined characteristics of a Btrieve file and statistics about the file's contents.

Format

`LOAD BUTIL –STAT` *`btrvFile`* `[/`**`O`***`owner`*`]`

btrvFile	The full pathname of the Btrieve file for which you want to display statistics.
owner	The owner name for the Btrieve file, if required.

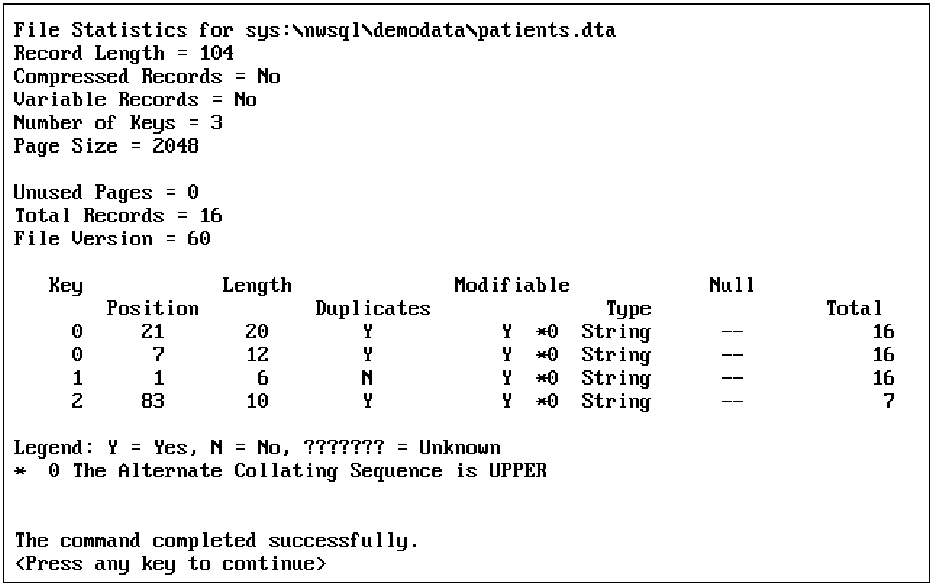
Example

The following example retrieves the file statistics for the PATIENTS.DTA file. The Btrieve file does not have an owner name.

```
load butil -stat sys:\system\515\patients.dta
```

The ouput screen resulting from this command looks similar to that shown in Figure 22.

Figure 22
Example of the Statistics Report Produced by the BUTIL -STAT Command



```
File Statistics for sys:\nwsql\demodata\patients.dta
Record Length = 104
Compressed Records = No
Variable Records = No
Number of Keys = 3
Page Size = 2048

Unused Pages = 0
Total Records = 16
File Version = 60

   Key            Length              Modifiable            Null
        Position          Duplicates               Type               Total
     0     21       20        Y           Y  *0  String      --           16
     0      7       12        Y           Y  *0  String      --           16
     1      1        6        N           Y  *0  String      --           16
     2     83       10        Y           Y  *0  String      --            7

Legend: Y = Yes, N = No, ??????? = Unknown
*  0 The Alternate Collating Sequence is UPPER


The command completed successfully.
<Press any key to continue>
```

This example shows that the file called PATIENTS.DTA was defined with a record length of 104 bytes, does not allow variable-length records, has 3 keys, and has a page size of 2,048 bytes. Sixteen records have been inserted into the file. The file does not use data compression and is using all its preallocated pages.

The Btrieve file version is 6.0. (If you created the Btrieve file with VATs, multiple ACSs, local-specific ACSs, or index balancing, the STAT command displays file version 6.1. Otherwise, it displays file version 6.0.)

Note

The STAT command designates case-insensitive keys and key segments with the letter I, descending keys with the symbol <, manual keys with the letter M, alternate collating sequence keys with an asterisk (*), and repeating duplicatable keys with the letter R. Indexes created with SINDEX are also designated with the letter R by default *unless* you specified the Reserved Duplicate Pointer element.

The remainder of the screen provides information about specific keys. For example, the screen shows that Key 0 allows duplicates, is modifiable, and consists of two segments:

- The first segment starts in position 21, is 20 characters long, allows duplicates, is modifiable, and will be sorted as a `string` type. The dashes indicate that a null value was not defined. The Total column indicates that 16 unique key values were inserted for this key.

- The second segment starts in position 7, is 12 characters long, allows duplicates, is modifiable, and will be sorted as a `string` type. Sixteen unique key values were inserted for this key.

Key 1 consists of one segment. It starts in position 1, is 6 characters long, does not allow duplicates, is modifiable, and will be sorted as a `string` type. Sixteen unique key values were inserted for this key.

Key 2 consists of one segment. It starts in position 83, is 10 characters long, allows duplicates, is modifiable, and will be sorted as a `string` type. Seven unique key values were inserted for this key.

Note

The STAT command handles indexes the same whether they were created by the Btrieve Create Supplemental Index operation (in Btrieve 6.*x*) or the Btrieve Create operation. The information displayed by the STAT command does not differentiate between these indexes.

VER

The VER command returns the version number of the Btrieve NLM loaded at the server.

Format

```
LOAD BUTIL –VER
```

Remarks

When you run the VER command, the utility displays messages similar to the following:

```
Btrieve Version is 6.1 NLM.
Operation completed successfully.
```

NetWare Btrieve Roll Forward Utility

The NetWare Btrieve Roll Forward utility is an interactive workstation utility that allows you to recover changes made to a Btrieve file between the time of the last backup and a system failure. The changes are stored in a log file.

If a system failure occurs, you can restore the backup copy of your Btrieve file and run the Roll Forward utility. The utility applies the changes stored in the log file to your backup copy.

This section discusses the following topics:

Using the Roll Forward Utility in Various Environments

Roll Forward utilities are available for DOS, OS/2, and MS Windows operating environments, as follows:

Checklist

❑ BROLLFWD.EXE—The Roll Forward utility for the DOS operating environment.

You run BROLLFWD from the command line. (The Roll Forward utility DBROLL.EXE, which was used in the DOS environment in previous releases, is obsolete.) For more information, see "Running the Roll Forward Utility in a DOS Environment" on page 178.

❑ PBROLL.EXE—The Roll Forward utility for the OS/2 environment.

You run PBROLL interactively.

❑ WBROLL.EXE—The Roll Forward utility for the MS Windows environment.

You run WBROLL interactively.

Note

The procedure for running PBROLL and WBROLL is the same. For more information, see "Running the Roll Forward Utility in an OS/2 or MS Windows Environment" on page 180.

Overview of the Recovery Process

In general, use the steps in the following procedure first to enable the NetWare Btrieve logging feature and then, in the event of a system failure, to recover your Btrieve files.

Procedure

1. **Back up your data files (and any current log files, if applicable), enable the logging feature using the Btrieve Setup utility, and create a log configuration file.**

   For more information, see "Setting Up Files for Logging" on page 174 and "Backing Up Data Files" on page 174, and "Example of Restoring a Data File" on page 195.

Warning

You cannot take advantage of the Roll Forward utility unless you have enabled the logging feature and created a log file *before* a system failure.

2. **Ensure that your system meets the system requirements for running the Roll Forward utility and then start the utility.**

   For more information, see “Running the Roll Forward Utility in a DOS Environment” on page 178 and “Running the Roll Forward Utility in an OS/2 or MS Windows Environment” on page 180.

3. **Set the appropriate Roll Forward utility options in one of the following ways:**

   - Activate the program features using the options from the Roll Forward utility’s Options menu (described on page 186).

   - Change the Roll Forward utility’s options manually by editing the initialization file (described on page 187).

   For more information, see “Setting Options for the Roll Forward Utility” on page 183.

4. **Select the Btrieve file (or files) you want to recover by adding that item to the Roll Forward utility’s queue.**

   For more information, see “Placing Items in the Queue” on page 187.

5. **Run the Roll Forward utility, which applies the changes stored in the log file to your backup copy of the Btrieve file.**

   For more information, see “Rolling Forward Items in the Queue” on page 192.

Note

For an example of the procedures used to handle data and log files in a roll-forward operation, see “Example of Restoring a Data File” on page 195.

Setting Up Files for Logging

To take advantage of NetWare Btrieve's logging feature and the Roll Forward utility, you must first set up your Btrieve files for logging, as described in the following procedure. Subsequent sections discuss each step in this process.

Procedure

1. **Back up your data files *before* you activate the logging feature.**

   For more information, see "Backing Up Data Files."

2. **Activate Btrieve's logging configuration option using the Setup utility (BSETUP.NLM).**

   For more information, see "Activating the Btrieve Logging Option" on page 175.

3. **Create the log configuration file, BLOG.CFG.**

   For more information, see "Creating the Log Configuration File" on page 176.

Backing Up Data Files

Be sure to make a backup copy of your Btrieve files before logging begins.

When you activate logging for a given file, NetWare Btrieve uses the corresponding log file to record all the operations that change the specified file. Btrieve continues appending subsequent operations to the end of this log file until you delete the log file. Consequently, it is important to perform periodic maintenance to reduce the size of your log files.

Important

Every time you back up your Btrieve files, delete the associated log files *before* executing any further operations that could change those log files. Synchronizing the backup data files and the associated log files is critical to recovering operations successfully.

Keep the original copy of the data files and their associated log files until you have successfully rolled the data files forward using the original logged information. Be aware that the Roll Forward operation modifies the log files after each roll forward attempt, so you cannot apply log entries twice to the same data files.

Be aware also that once you have performed a Roll Forward operation, Btrieve does not retain the original logging information. Instead, if you have enabled the logging feature, Btrieve adds to the log file a record of all the operations that affect the specified file subsequent to the last Roll Forward operation.

For an example of the procedures used to handle data and log files in a roll-forward operation, see “Example of Restoring a Data File” on page 195.

Activating the Btrieve Logging Option

You can activate NetWare Btrieve’s logging feature by specifying Yes for the Logging of Selected Files configuration option in the Setup utility. The default setting for this option is No.

If you did not specify Yes for this option when you configured NetWare Btrieve, complete the following steps to activate the Btrieve logging feature:

Procedure

1. **Run the Setup utility (BSETUP.NLM).**

2. **When the Available Options menu appears, select Set Btrieve Configuration.**

3. **When the Current Btrieve Configuration screen appears, specify Yes for the Logging of Selected Files option.**

4. **Press Escape.**

5. **When the Save Configuration Changes? window appears, select Yes.**

6. **To have your changes take effect, unload Btrieve using the BSTOP command and then reload it using the BSTART command.**

   Btrieve reloads with the logging feature activated.

For additional information about this procedure see Logging of Selected Files on page 69 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Creating the Log Configuration File

The name of the NetWare Btrieve log configuration files is BLOG.CFG. It specifies the names of all the Btrieve files for which you want to log changes on a given volume.

You should create a BLOG directory at the root of each volume containing Btrieve files for which you want to log changes. You can then create a BLOG.CFG file in each BLOG directory and place entries in it, as follows:

Procedure

1. **Create an empty BLOG.CFG file.**

   You can use any text editor to create the BLOG.CFG file. Remember that you must create a BLOG.CFG file in the BLOG directory of each volume on which you want to log files.

2. **Open the BLOG.CFG file.**

3. **Create an entry in the BLOG.CFG file for each Btrieve file for which you want to log operations.**

Use the following format to create the BLOG.CFG file entries:

```
\directory1\btrvFile[=\directory2\logFile]
```

directory1 The path to the Btrieve file to be logged.

Do not include server names, volume names, or DOS drive letters.

btrvFile The name of the Btrieve file to be logged.

directory2 The path to the log file.

If the log file and the Btrieve file are on the same volume, you can omit the server and volume names. If they are on different volumes, you must include the server and volume names.

When including the server name, place a double backslash (\\) before it.

logFile	The name of the log file. Although the log file and the Btrieve file can be on different volumes, they cannot be on different servers.

Make sure each entry fits completely on one line. (Each line can contain a maximum of 256 characters.) If you have room, you can place multiple entries on the same line, but you must separate each entry with at least one space.

Note A single log entry cannot contain any spaces, although you must separate multiple entries on the same line with at least one space.

If you do not provide a name for the log file, Btrieve assigns the original filename plus a .LOG extension to the file when you first open it.

For example, assume that you did not specify a name for log file associated with the Btrieve file TEST01.DAT located in the directory TEST. Btrieve would assign the full name \TEST\TEST01.LOG to the default log file, which in this case, shares the same directory as the Btrieve file.

The following three examples show sample entries in the file \BLOG\BLOG.CFG on the SYS: volume of the CORP server. Each entry produces the same result: activity in the file \DATA\B.BTR on the CORP server’s SYS: volume is logged into the file \DATA\B.LOG on the CORP server’s SYS: volume.

```
\data\b.btr

\data\b.btr=\data\b.log

\data\b.btr=\\corp\sys:\data\b.log
```

The next example (again, a sample entry in \BLOG\BLOG.CFG on the CORP server's SYS: volume) shows how to log activity to a volume other than the Btrieve data file's volume:

```
\data\b.btr=\\corp\vol1:\data\b.log
```

This entry directs Btrieve to log activity in the file \DATA\B.BTR on the CORP server's SYS: volume into the log file \DATA\B.LOG on the VOL1: volume of the CORP server. You can also use the following syntax to achieve the same result:

```
\data\b.btr=\\corp\vol1:data\b.log
```

Running the Roll Forward Utility in a DOS Environment

To run the Roll Forward utility in a DOS environment, enter the BROLLFWD command using the following format:

```
BROLLFWD <btrvFile | @listFile | /A>
  [/D:nn] [/T:nn] [/K:nn] [/H] [/V] [/L]
  [/O:ownerName]
```

The following list describes the BROLLFWD command syntax:

btrvFile Specifies the name of a single Btrieve file to be recovered.

@listFile Specifies the name of a text file that contains a list of Btrieve filenames separated by one or more spaces. Use a list file to recover multiple files.

/A Specifies that you want to recover all the Btrieve files in the BLOG.CFG file.

/D: Specifies the data buffer size (in kilobytes) that the Roll Forward utility allocates for Btrieve log operations. /D: is optional.

The default size is 8 KB, the minimum is 1 KB, and the maximum is 64 KB. You can specify the length in increments of 1 KB.

/T:

Specifies the length of the data (in bytes) that will be printed in the list file for each operation that is rolled forward. /T: is optional.

Valid data lengths range from 1 through the value of the data buffer size specified with the /D: option. The default value is 40 bytes.

/K:

Specifies the length of the key (in bytes) that will be printed in the list file for each operation that is rolled forward. /K: is optional.

Valid lengths for printing keys range from 1 through 255 bytes. The default value is 10 bytes.

/H

Specifies that the Btrieve operations in the list file will be printed in hexadecimal format. The default prints the data and key in decimal numbers. /H is optional.

/V

Specifies that for each logged file in the list file, the utility will add the time stamps of the Roll Forward operation and log file creation. /V is optional.

For each logged operation, it adds the name of the user who performed the operation, the internetwork address of the source workstation, the time stamp indicating when the operation was performed, and the record length and key number used in the operation.

/L

Specifies that you want only to *list* the logged operations. The logged operations will not be executed. The operations will be listed to the standard output device. /L is optional.

/O:

Specifies an owner name. If the backup copy of the Btrieve file you want to recover has a Btrieve owner name, you must provide this option. This protects the owned files from being changed inadvertently.

Typically, all owned files in an application have the same owner name. Therefore, the utility assumes that all Btrieve files listed in the file list have the same owner name. However, some Btrieve files in a file list may have different owners.

If a Btrieve file has an owner name, that file has only one owner name. In that case, the utility prompts you to enter the owner name.

Similarly, if you do not specify /O and the utility encounters a Btrieve file that requires an owner name, BROLLFWD prompts you for that owner name.

ownerName Specifies the owner name of the Btrieve files to be accessed. When you use /O, you must specify an owner name.

Running the Roll Forward Utility in an OS/2 or MS Windows Environment

The following list shows a few ways you can run the Roll Forward utility:

From This Position	Do This
OS/2 command line	Enter the following command: pbroll
Presentation Manager	Select the Roll Forward icon by double-clicking on it.
MS Windows	Double-click on the Roll Forward icon, or choose Run from the File pulldown menu and enter the following command: wbroll

Note The following information applies to both OS/2 and MS Windows operating environments, but the screen examples show MS Windows screens only.

To use the Roll Forward utility in the OS/2 and MS Windows environments, complete the following steps:

Procedure

1. **Back up your data and log files.**

   For more information, see “Backing Up Data Files” on page 174.

2. **Set the Roll Forward utility's program options.**

   For more information, see "Setting Options for the Roll Forward Utility" on page 183.

3. **Place in the utility's queue all the items you intend to roll forward.**

   For more information, see "Placing Items in the Queue" on page 187.

4. **Start rolling forward the items in the queue.**

   For more information, see "Rolling Forward Items in the Queue" on page 192.

Following is a brief description of the Roll Forward utility's pulldown menus. Subsequent sections describe each step in the Roll Forward recovery procedure in detail.

Using the Roll Forward Pulldown Menus

After starting the Roll Forward utility, you can access two pulldown menus: Queue and Options.

Hint

If you are not using a mouse, you can access the menus by pressing and holding the Alt key while typing the letter highlighted in the menu selection. For example, to select the Queue pulldown menu, hold down the Alt key and press Q. To move between fields in the dialog boxes, use the Tab key.

Queue Menu

When you select Queue from the main menu, a pulldown menu offers the following:

Add — Displays a dialog box in which you can specify items to be placed in the queue.

View — Displays a dialog box in which you can view the queued items. If no items are in the queue, this selection is disabled.

Start	Begins the process of rolling forward all items in the queue. Like the View selection, this selection is disabled if no items are in the queue.
Exit	Exits the utility. In the MS Windows and OS/2 environments, you can also press F3 to exit.

Options Menu

When you select Options from the main menu, a pulldown menu offers the following:

Options	Displays a dialog box that enables you to set the data buffer length and the list options.
About	Displays the version of the Roll Forward utility that you are currently running.

Setting Options for the Roll Forward Utility

Set the desired program options for the Roll Forward utility before using the utility to apply the logged changes to the specified Btrieve file or files. These options control the following:

- The size of the data buffer used to retrieve records
- Whether the Roll Forward procedure is an exclusive operation
- The contents and format of the list file (BROLL.LST)

Table 7 describes each option in detail. Subsequent sections describe the two methods you can use to set these options:

- "Setting Options from the Options Menu" on page 186
- "Setting Options Manually in the MS Windows Initialization File" on page 187. (You cannot edit the OS/2 initialization file.)

Table 7
Roll Forward Options

Program Option	Description
Data Length	Specifies the number of kilobytes allocated for the data buffer that the utility uses to process the logged entries. This number should be at least as large as the largest record to be rolled forward. The default is 4 KB.

Table 7 *continued*
Roll Forward Options

Program Option	Description	
Exclusive Operation	Runs in one of two ways, depending on your operating system:	
	MS Windows	MS Windows 3.*x* emulates multitasking and lets you run more than one application concurrently. If you select Exclusive Operation, the Roll Forward utility uses the CPU time exclusively. If you do not select this option, the utility shares CPU time with other applications. If you plan to run recorder-type programs or batch execution programs with Roll Forward, check this box to ensure correct operation. Selecting Exclusive operation also enhances performance slightly.
	OS/2	OS/2 provides true multitasking; the Roll Forward utility can always run concurrently with other applications. You can, however, vary the priority of the Roll Forward thread to accommodate other threads that are running. The following priorities are available: Idle—Runs only when no other tasks are waiting for the CPU Low—Lower priority than normal Normal—The default thread priority High—Higher priority than normal

Table 7 *continued*

Roll Forward Options

Program Option	Description	
List Files	Specifies the listing options for the list file BROLL.LST. You can select one of the following:	
	Verbose	For each logged file, this option adds the time stamps of the Roll Forward operation and log file creation to the list file. For each logged operation, it adds the name of the user who performed the operation, the internetwork address of the source workstation, the time stamp indicating when the operation was performed, and the user-defined lengths of data and key numbers used in the operation.
	Data to list (bytes)	Specifies the length of the data buffer that will be printed in the list file for each operation that is rolled forward.
	Key to list (bytes)	Specifies the length of the key buffer that will be printed in the list file for each operation that is rolled forward.
	ASCII	Lists the Btrieve operation values in ASCII mode.
	Hex	Lists the Btrieve operation values in hexadecimal mode.

Setting Options from the Options Menu

To set Roll Forward options using the Options pulldown menu, complete the following steps:

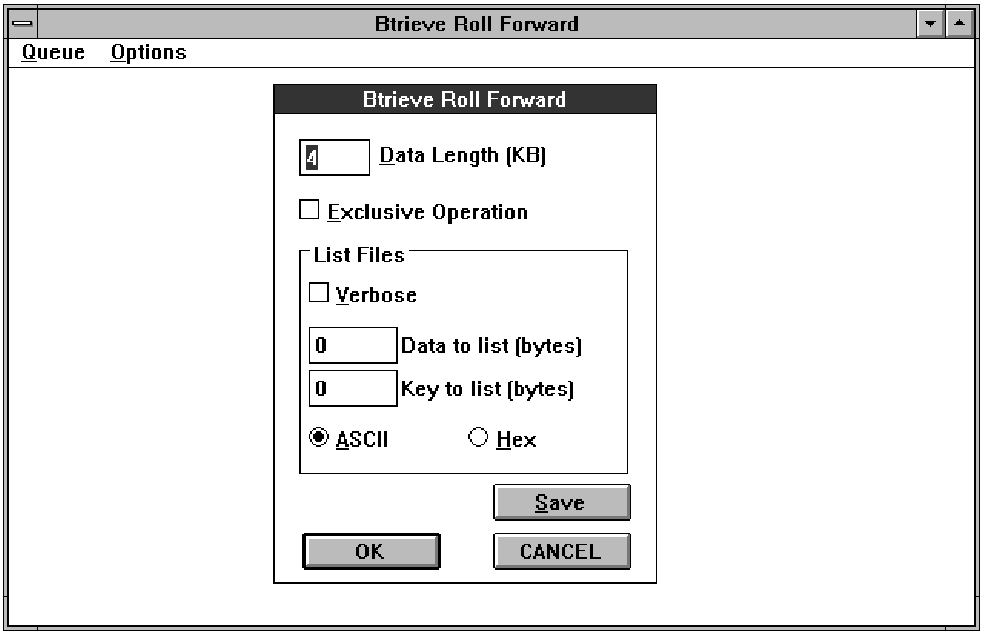
Procedure 23

1. **Select Options from the Roll Forward main menu.**

2. **Select Options from the Options pulldown menu to display the Btrieve Roll Forward dialog box, shown in Figure 23.**

Figure 23

Roll Forward Utility: Options Dialog Box



4 Data Length (KB)

□ Exclusive Operation

□ Verbose

0 Data to list (bytes)

0 Key to list (bytes)

● ASCII ○ Hex

3. **Using the guidelines provided in Table 7 on page 183, set the options you want to activate for logging your selected files.**

4. **After setting the options, select one of the following buttons to execute your choice:**

   - Save—Accepts and saves the changes you have made to the NOVDB.INI file.

- OK—Accepts the changes but does not save them to the NOVDB.INI file.
- CANCEL—Cancels the changes and returns to the previous screen.

Setting Options Manually in the MS Windows Initialization File

You can also change the setting of the Roll Forward utility's options by manually editing the MS Windows initialization file, NOVDB.INI. (You cannot edit the OS/2 initialization file).

These settings are specified under `[wbroll]` in NOVDB.INI. The following is an example specification for `[wbroll]`:

```
[wbroll]
datalength=4
exclusive=no
outputmode=ASCII
listverbose=yes
datalist=32
keylist=16
```

See Table 7, " Roll Forward Options," on page 183 for an explanation of these options.

Placing Items in the Queue

The Roll Forward utility works on a queued-job basis. When you specify the Btrieve files that are to be rolled forward, the utility places them in the queue.

This section discusses the Roll Forward utility's queue and explains how to perform the following tasks:

- Adding Items to the Queue
- Deleting Items from the Queue
- Changing List Options for a Queued Item
- Viewing Items in the Queue

Adding Items to the Queue

The queue can hold a maximum of 32 items. Any of the following represents one item:

- An individual Btrieve file
- A text file listing several Btrieve files
- All files from a specified volume

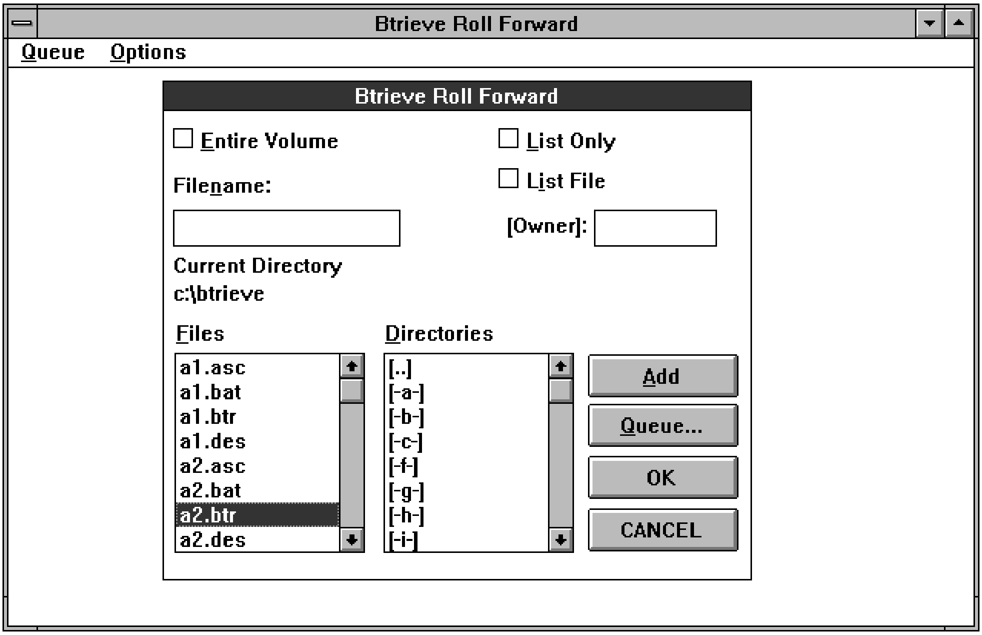
To add items to the queue, complete the following steps:

Procedure 23

1. **Select Add from the Queue pulldown menu.**

   The Add dialog box, similar to that shown in Figure 24, appears:

Figure 24
Roll Forward Utility: Add File Dialog Box



2. Select the Btrieve file or files to be rolled forward, as follows:

- To select all the files from a specified volume, select the Entire Volume check box.
- To select a particular file, scroll the Directories list box (the list box on the right) to find your directory or drive. Select the directory name and then choose the filename from the Files list box (the list box on the left).

Note Setting `show dots=on` in your NET.CFG file makes the . and .. directory notation visible, allowing you to select the previous directory or root directory from the Directories list box. Refer to your NetWare documentation for details about modifying the NET.CFG file.

- To select a text file containing a list of files, enter the filename in the Filename text box.
- To select all available files with a certain extension, enter a wildcard character for the filename, followed by the extension, and press Enter.

3. Specify the list option for the queue item you are adding, as follows:

- List Only —If you want only to list the Btrieve operations to be rolled forward *without* actually performing the procedure, select *both* the List Only *and* List File check boxes. Btrieve then lists the operations in the BROLL.LST file in the BLOG directory.
- Roll Forward Only—If you do not want to list the operations that are rolled forward but you do want to perform the roll forward procedure itself, *do not select either* the List Only or List File check box; leave them both unchecked.
- Both List and Roll Forward—If you want to list the operations in BROLL.LST *and* perform the roll forward procedure, select *only* the List File check box.

4. If the Btrieve file has an owner name, specify the owner name in the optional [Owner] text box.

5. Select the Add button to add the item to the queue.

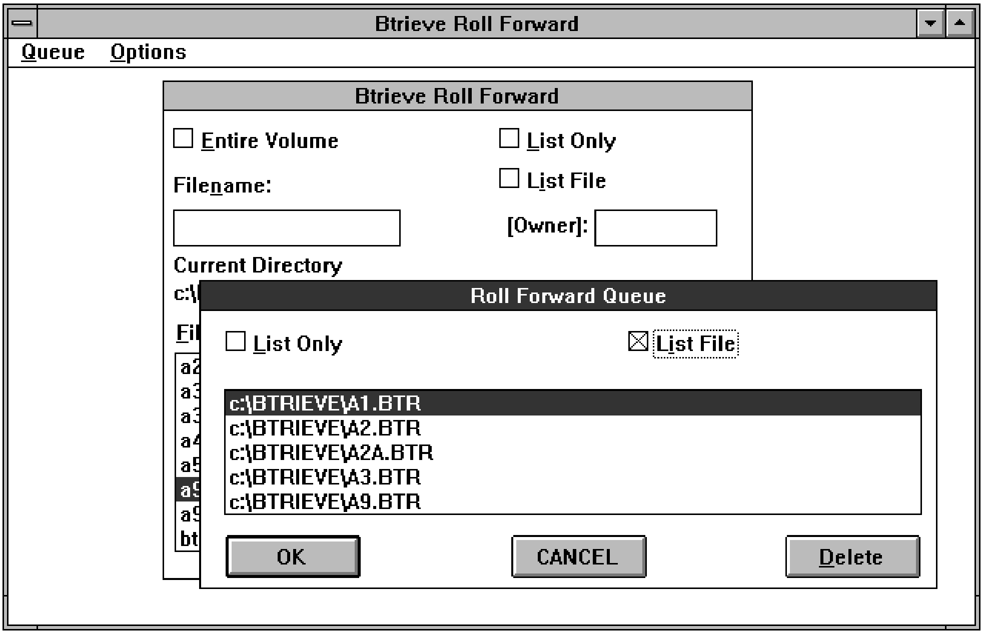
Notice that the Queue button, which has been dimmed until now, becomes available after you select the Add button.

6. **Repeat Steps 2 through 5 to add each additional item to the queue.**

7. **To review the items you have placed in the queue, select the Queue button.**

   The items selected appear on a screen similar to that shown in Figure 25.

Figure 25
Roll Forward Utility: Items in the Queue List



c:\BTRIEVE\A1.BTR
c:\BTRIEVE\A2.BTR
c:\BTRIEVE\A2A.BTR
c:\BTRIEVE\A3.BTR
c:\BTRIEVE\A9.BTR

8. **When you are finished, click on the OK button.**

Note

At any time, you can click on the CANCEL button to cancel your changes and return to the previous screen.

Deleting Items from the Queue

If you need to delete an item from the queue, complete the following steps:

Procedure

1. **Select View from the Queue pulldown menu.**

2. **Select the item you want to delete.**

3. **Click on the Delete button to remove the item from the queue.**

4. **Click on the OK button.**

Note

If you change your mind and want to cancel your deletion, click on the CANCEL button instead of OK.

Changing List Options for a Queued Item

You can use either of the following methods to change the list options (that is, to change your choices regarding the List Only and List File check boxes) for a given queue item:

- Select Add from the Queue menu and then select the Add button. Next, select the relevant item and choose the list option you prefer.

- Select View from the Queue menu, select the relevant item, and choose the list option you prefer.

Viewing Items in the Queue

You can use either of the following methods to view items in the queue:

- From the Queue pulldown menu, select View to display a dialog box that lists the queued items. (You can select this option only if the queue has one or more items in it.) The View dialog box, shown in Figure 26, appears.

- While you are adding items to the queue, click on the Queue button to list the files in the queue.

Rolling Forward Items in the Queue

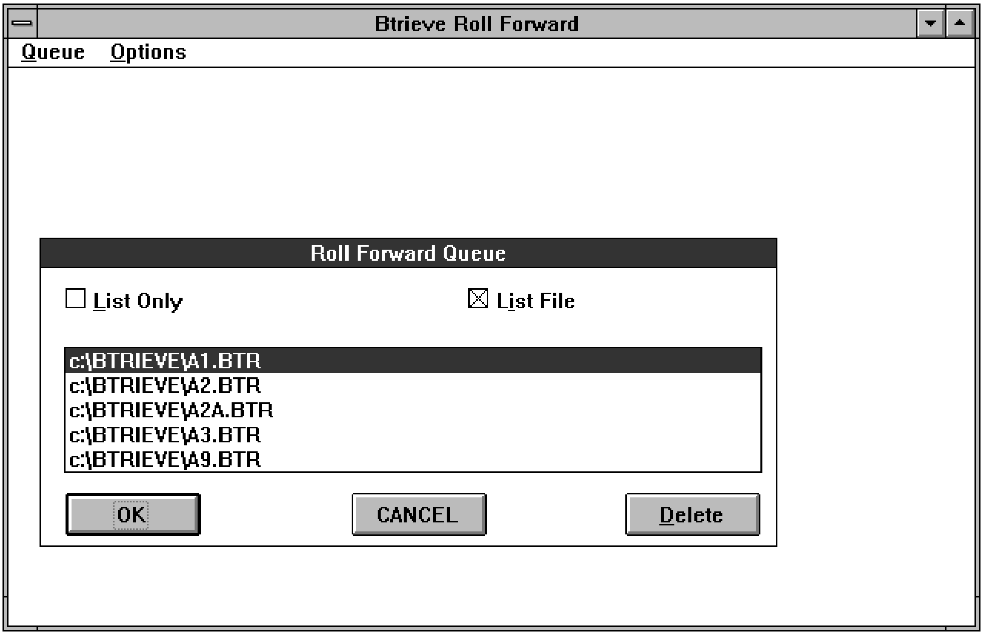
Once the queue contains all the items for which you want to roll forward changes, you are ready to start the roll forward procedure. Before you begin, however, back up your data and log files, as explained in “Backing Up Data Files” on page 174.

Procedure

1. **Select Start from the Queue pulldown menu.**

   The Queue Menu, with the Start command highlighted, appears.

Figure 26
Roll Forward Utility:
View Queue Contents Dialog Box



☐ List Only
☒ List File

c:\BTRIEVE\A1.BTR
c:\BTRIEVE\A2.BTR
c:\BTRIEVE\A2A.BTR
c:\BTRIEVE\A3.BTR
c:\BTRIEVE\A9.BTR

Note

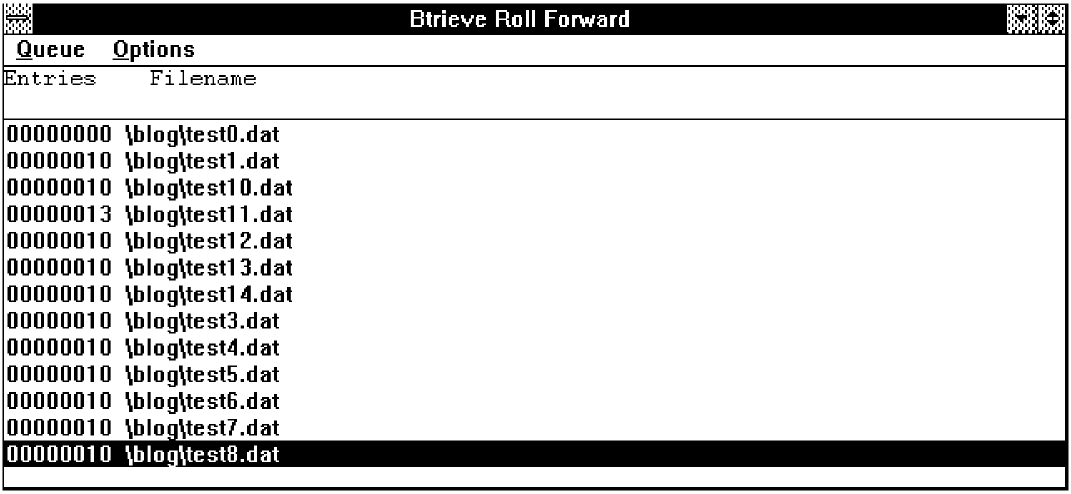
The Roll Forward utility allows a maximum of 250 concurrent transactions per Btrieve file during the roll forward process.

2. **After you select Start, the utility lists each Btrieve file being rolled forward and specifies the number of logged entries for each file.**

   The Roll Forward utility displays a screen similar to that shown in Figure 27. The number of logged entries is shown to the left of the filename.

Figure 27

Roll Forward Utility: Roll Forward List Screen



```
Entries     Filename

00000000  \blog\test0.dat
00000010  \blog\test1.dat
00000010  \blog\test10.dat
00000013  \blog\test11.dat
00000010  \blog\test12.dat
00000010  \blog\test13.dat
00000010  \blog\test14.dat
00000010  \blog\test3.dat
00000010  \blog\test4.dat
00000010  \blog\test5.dat
00000010  \blog\test6.dat
00000010  \blog\test7.dat
00000010  \blog\test8.dat
```

Special Considerations When Using Logging and the Roll Forward Utility

When using NetWare Btrieve's logging feature and the Roll Forward utility, consider the following:

- NetWare Btrieve does not allow a log file to contain log entries created by different versions of NetWare Btrieve. That is, all entries in a log file must be logged under the same version of NetWare Btrieve. Otherwise, NetWare Btrieve displays a system error message on the server console and ignores logging for the specified Btrieve file.

- If you need to restart NetWare Btrieve, specify the same index balancing setting that was used when you first loaded NetWare Btrieve (after a backup) and performed logging. Otherwise, you may receive a Status Code 43 (The specified record is invalid) when you run the Roll Forward utility.

- If you need to run the Roll Forward utility (for example, after a system crash), load the same version of Btrieve and specify the same index balancing setting that was used during logging. Otherwise, you may receive a Status Code 43 (The specified record is invalid).

Note

If you want to switch to a different version of NetWare Btrieve, or you want to change the index balancing setting, first create a backup of the Btrieve files to be logged and then delete the corresponding log files.

- If you attempt to create a log file for a Btrieve file that contains records larger than 57 KB, you may receive a Status Code 43 (The specified record is invalid) when you run the Roll Forward utility.

Example of Restoring a Data File

This section includes instructions for handling data and log files so you can restore a lost or corrupted logged Btrieve file. It includes the following subsections:

- "Making a Backup Copy and Enabling Logging"
- "Restoring a Logged File from a Backup Copy" on page 196

Making a Backup Copy and Enabling Logging

The following procedure explains how to handle a Btrieve file called X.BTR to prepare it for logging.

Procedure

1. **Make a copy of the Btrieve file X.BTR called (for example) X.BAK.**

   For additional information about this process, see the section "Backing Up Data Files" on page 174.

2. **Turn on the logging feature.**

   For additional information about this process, see the section "Activating the Btrieve Logging Option" on page 175.

After you have enabled logging and restarted Btrieve, Btrieve writes operations to both the data file (X.BTR) and to the log file, which is called X.LOG.

Note

To keep X.LOG from becoming too large, you may periodically want to make a new backup copy of the logged Btrieve file. Then delete the existing X.LOG file because the new copy of X.BAK already contains all the changes to the X.BTR file that were logged in the existing copy of X.LOG. If logging is enabled, Btrieve will create a new copy of X.LOG when you next work with the X.BTR file.

Restoring a Logged File from a Backup Copy

If you lose a Btrieve file or if, for some reason, a data file becomes corrupted, follow this procedure to restore that file:

1. **Copy the backup file, X.BAK, to the Btrieve file (X.BTR).**

2. **Copy the logging file, X.LOG, to a new file called (for example) X.LLL.**

3. **Use the Roll Forward utility to apply the operations logged in X.LOG to X.BTR.**

   For detailed information about this process, see the section “Rolling Forward Items in the Queue” on page 192.

   If the roll forward operation is successful, proceed to Step 4. If for some reason the first roll forward procedure does not complete successfully, proceed to Step 5.

   Note: One reason the roll forward process might fail is that Btrieve is configured differently during a roll forward operation than it is during the original logged operations.

4. **If the roll forward operation completes successfully, follow these steps:**

   4a. Copy X.BTR to X.BAK to make a new backup copy of the restored Btrieve file.

   4b. Delete X.LOG and X.LLL.

5. **If the roll forward operation did not complete successfully, follow these steps:**

   5a. Copy X.BAK to X.BTR.

   5b. Copy X.LLL to X.LOG

   5c. Perform the roll forward operation again.

   5d. Make a new backup copy of the data file and delete the old log file and X.LLL.

appendix A Btrieve Concepts

This appendix describes the fundamental concepts behind the management of Btrieve files, records, keys, and indexes. It includes the following sections:

Overview of How Btrieve Works

The Btrieve product stores information in Btrieve data files. Inside each data file is a collection of records (described in detail in “Records” on page 207).

A record contains bytes of data. That data might represent an employees’s name, ID, address, phone number, rate of pay, and so on.

For example, when an application retrieves the record for Cliff Jones, the information in the record might be displayed as follows:

```
Jones Cliff D 340873 2341 Baxter Austin TX 512-555-
2345 2146.00
```

Note

The application, not Btrieve, is responsible for the actual appearance of the data in a record—that is, the format of the data on screen or in a report.

Btrieve sees this record (or any record) only as a collection of bytes; it does not recognize logically discrete pieces of information within a record. To Btrieve, no last name, first name, employee ID, and so on exist inside a record. The record is simply a collection of bytes.

For example, an application that inserts information into or retrieves information about Cliff Jones from the Btrieve file might specify that the fields in the record use a data structure based on the format shown in Table 8.

Table 8
Example Data Structure for a Sample Record

Information in Record	Length in Bytes	Data Type
Last name	25	Null-terminated string
First name	25	Null-terminated string
Middle initial	1	Char (byte)
Employee ID	4	Long (4-byte integer)
Street address	30	Null-terminated string
City	25	Null-terminated string
State	11	Null-terminated string

Table 8 *continued*

Example Data Structure for a Sample Record

Information in Record	Length in Bytes	Data Type
Phone number	12	Null-terminated string
Pay rate per month	4	Long (4-byte integer)

Inside the Btrieve file, however, this information comprising Cliff Jones' record would be stored as a collection of bytes.

The only discrete portions of information that Btrieve can recognize in a Btrieve file are *keys* (described in detail in "Keys" on page 211). An application (or user) can designate any collection of bytes in a record as a key.

Using Btrieve sorts records on the basis of the *values* in any specified key. Btrieve can also find a particular record based on a specified key value.

In the example shown in Table 8, the 25 bytes that contain the last name in each record might be designated as a key in the Btrieve file. The sample application could use the last name key to obtain a listing of all the employees named Smith, or it could obtain a listing of all employees and then display that listing, sorted by last name.

Keys also allow Btrieve to access information quickly. For each key defined in a Btrieve data file, Btrieve builds an *index* (described in detail on page 231.) The index is stored inside the Btrieve data file itself and contains a collection of pointers (addresses or offsets) into the actual data within that file. A key value is associated with each pointer.

In the example shown in Table 8, the key "last name" has an index. Inside that index is a collection of last names: one last name for every employee. For every last name in the index, a pointer indicates where the information about that employee is located in the Btrieve data file.

Normally, when accessing or sorting information for an application, Btrieve does not search through all the data in its data file. Instead, it goes to the index, performs the search, and then manipulates only those records that meet the application's request.

Btrieve Files

A *file* is the highest-level database entity you can access using Btrieve. Btrieve allows a maximum file size of approximately four gigabytes. You can create Btrieve data files and define their characteristics by using either of the following:

- The Btrieve Maintenance utility commands CREATE, CLONE, and so on (described in "NetWare Btrieve Maintenance Utility Commands" on page 143 in Chapter 5, "Using NetWare Btrieve Utilities")
- The Btrieve Create (14) operation in an application

This section discusses the following information about files:

- "File Components" on page 200
- "File Types" on page 203
- "File Space Allocation" on page 205
- "Special File Considerations" on page 206

File Components

Btrieve files consist of a series of *pages*. A page is the unit of storage that Btrieve transfers between memory and disk. You specify a fixed size for each page when you create a file. The page size is always a multiple of 512 bytes, up to 4,096 bytes.

As discussed in the following sections, a Btrieve file is composed of these types of pages:

- "FCR Pages" on page 201
- "Page Allocation Table Pages" on page 201
- "Data Pages and Variable Pages" on page 202
- "Index Pages" on page 202
- "Alternate Collating Sequence Page" on page 203

FCR Pages

The first two pages in every Btrieve 6.*x* file are the File Record Control (FCR) pages. At any given time, Btrieve considers one of the FCR pages to be *current*. The current FCR page contains the latest information about the file, such as the file size, page size, alternate collating sequence (ACS) name (if any), and other characteristics of the file.

Page Allocation Table Pages

Page Allocation Table (PAT) pages are part of Btrieve 6.*x*'s internal implementation for converting between physical and logical page numbers in a Btrieve file.

Btrieve knows the specified page length of every Btrieve file. Therefore, it can determine the location of each page by knowing that page's physical page number. For example, if the page size of a Btrieve file is 4,096 bytes, page 0 starts at offset 0, page 1 starts at offset 0x1000 (4096), page 2 at offset 0x2000 (8192), and so on.

Since this tracking deals with a page's physical location within the Btrieve file, these page numbers are called *physical page numbers*.

However, as records are modified in a Btrieve file (or as indexes are created or deleted), information changes location within the file. For example, a record describing employee Cliff Jones might be on physical page 34 when you begin an update, but on physical page 38 when you finish the update.

To keep track of information as it moves around the Btrieve file, Btrieve uses the concept of *logical page numbers*. Records always reside on the same logical page, even though they move from one physical page to another as modifications occur.

The PAT tracks the mapping of logical page numbers to physical page numbers.

Data Pages and Variable Pages

When your application inserts a record into a file, Btrieve stores that record on either a data page or on both a data page and a variable page. Data pages contain fixed-length records; variable pages contain variable-length records. (Records are discussed in detail on page 207.)

If a file does not allow variable-length records (described on page 207) or data compression (described on page 263), it will have data pages but no variable pages.

Each data page may contain one or more data records. The number of records per page depends on the record length. Btrieve does not split the fixed-length portion of a record across two data pages.

If a file allows variable-length records or data compression (or both), it will contain both data pages and variable pages. The data pages contain only the fixed-length portions of the records. The variable pages contain only the variable-length portions of the records.

If the variable-length portion of a record is longer than the defined page size (minus overhead) for the file, Btrieve splits the variable-length portion over multiple variable pages.

Index Pages

Index pages are nodes in a B-tree structure. Each node contains key values for the data records. Generally, index pages contain many different key values.

Each key value on the page has a record address. Btrieve uses this address to retrieve records containing a specified key value.

Note

When you enable linked-duplicatable keys (discussed on page 214), each key value on the page has two record addresses that Btrieve uses when retrieving records.

Alternate Collating Sequence Page

If any index in a Btrieve file uses an alternate collating sequence (ACS), the file will have at least one ACS page. (ACSs are described in detail on page 216.)

File Types

Btrieve allows you to create three different types of files:

- ◆ Standard Btrieve Files
- ◆ Data-Only Files
- ◆ Key-Only Files

Standard Btrieve Files

A standard Btrieve 6.*x* file contains two FCR pages followed by a number of PAT pages, index pages, data pages, and possibly variable pages. As this implies, you can create a standard Btrieve file for use with either fixed- or variable-length records.

Because standard Btrieve files contain all the index structures and data records, Btrieve can dynamically maintain all the index information for the records in the file. You can use any of the Btrieve record retrieval operations to access the information stored in a standard Btrieve file.

Data-Only Files

Btrieve also lets you create data-only files that hold only data. When you create a data-only file, you do not specify any key information, and Btrieve allocates no index pages for the file (only the two FCR pages and the PAT pages). This results in a smaller initial file size than for standard Btrieve files.

When an application inserts records into the file, Btrieve stores them in the chronological order of insertion.

Important

The chronological ordering of records can change when you delete records and insert new ones, or when the Btrieve file is rebuilt. An application should not depend on the chronological ordering of records in a Btrieve file.

Btrieve does not maintain or create any index pages as the records are inserted. At this point, an application can access the records using only the Step operations or the Get Direct/Record (23) operation, all of which use the physical location to find records.

At any time after creating a data-only file, you can add an index using the Btrieve Maintenance utility command SINDEX, or an application can issue a Create Index (31) operation. Then, the new index can be used to retrieve records with the Get operations.

Key-Only Files

Key-only files contain only FCR pages followed by a number of PAT pages and index pages. If a key-only file has an ACS, it may also have an ACS page. Similarly, if anyone has used the Scalable SQL relational data access system to define referential integrity (RI) constraints on the file, the file may contain one or more variable pages.

In a key-only file, the entire record is stored with the key, so no data pages are required. Key-only files are useful when your records contain a single key, and that key takes up most of each record. Another common use for a key-only file is as an external index for a standard Btrieve file.

The following restrictions apply to key-only files:

- Each file can contain only a single key.
- The maximum record length you can define is 253 bytes (or 255 bytes for a pre 6.*x* format file).
- Key-only files do not allow data compression.

File Space Allocation

Btrieve automatically reallocates file space when you insert or delete Btrieve records. The following paragraphs explain how Btrieve dynamically allocates space and uses free space.

Dynamic Expansion

Btrieve allocates disk space as needed. If there is not enough room in the file when an application inserts new records, Btrieve dynamically allocates additional data and index pages to the file. The size of each allocated page is the same as the page size with which the file was created.

Btrieve also updates directory structures to reflect the new file size.

Once Btrieve allocates space to a file, that space remains allocated as long as the file exists. To reduce the space required for a file from which numerous records have been deleted, you can create a new file with the same characteristics as the original file using the Maintenance utility command CLONE (described on page 144).

Then use one of the following Maintenance utility commands or command sequences to read the records from the original file and insert them into the new file:

- COPY
- RECOVER, LOAD
- SAVE, LOAD

For detailed information about using these commands, refer to "NetWare Btrieve Maintenance Utility Commands" on page 143 in Chapter 5, "Using NetWare Btrieve Utilities."

Free Space Utilization

When you delete a record, the disk space it formerly occupied is put on a free space list. When an application inserts new records, Btrieve uses the free space instead of allocating additional pages to the file. Btrieve's method of reusing free space eliminates the need to reorganize files to reclaim disk space.

Special File Considerations

The following sections describe types of files that require special consideration when used with different versions of Btrieve:

- *Transactional files* are files created or flagged transactional using the NetWare® Transaction Tracking System (TTS) protection feature.
- *Pre-image files* are files created by Btrieve for use with versions of Btrieve files prior to 6.0.

Transactional Files

NetWare Btrieve uses the NetWare TTS feature to ensure the physical integrity of a file. When you use TTS to guard your files, your files have the transactional attribute set and are called transactional files.

To set up your files as transactional, set a transactional bit in NetWare with the NetWare FLAG command, which flags an existing Btrieve file (or files) as transactional. A file can also be flagged transactional at the time of its creation is the proper parameter was set when Btrieve was loaded.

For information on NetWare TTS or the FLAG command, refer to your NetWare documentation.

Btrieve 6.*x* and Pre-image Files

Btrieve 6.*x* creates a pre-image file when writing to a Btrieve file that was created in either one of two ways:

- With a pre-6.*x* version of Btrieve
- With Btrieve 6.*x* if the 6.*x* version of Btrieve was loaded with the –D option (thereby forcing all files created to use the pre-6.*x* format)

However, the format of a pre-image file created by Btrieve 6.*x* differs from the format of a pre-image file created by pre-6.*x* versions of Btrieve. This means that the earlier versions of Btrieve will not be able to accurately interpret a pre-image file created by Btrieve 6.*x*. Also, Btrieve 6.*x* will not be able to accurately interpret a pre-image file created with an earlier version of Btrieve.

If you need to change from running Btrieve 5.*x* to running Btrieve 6.*x* (or vice versa) and your application will be accessing pre-6.*x* Btrieve files, you must delete all pre-image files prior to changing Btrieve versions.

For detailed information about this process, see “Checking for and Removing Extraneous Pre-Image Files” on page 49 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

For additional information about pre-image files, see “Pre-imaging” on page 246 in this appendix.

Records

A *record* represents a set of logically associated data items in a Btrieve file. Generally, a record is the unit transferred between an application and Btrieve in a single operation.

No inherent restrictions apply to the number of records allowed in a Btrieve file. A record can consist entirely of a fixed-length portion (a fixed-length record), or it can consist of a fixed-length portion followed by a variable-length portion (a variable-length record). All the keys defined for the file must be located within the fixed-length portion of the record.

Fixed-Length Records

In a file that uses fixed-length records, the length of each record in that file must be the same. The maximum length of a fixed-length record depends on the physical page size and the number of duplicate keys you define for the file.

Btrieve allows a fixed-length record to contain up to 4,088 bytes (4,096 minus 6 bytes for page overhead information, and 2 bytes for the record usage count). Therefore, the largest possible fixed-length record is 4,088 bytes.

Variable-Length Records

Variable-length records store data of indeterminate length. When you create a Btrieve file, you can specify that you want the file to use variable-length records.

A variable-length record has a fixed-length portion and a variable-length portion. While the fixed-length portion of all variable-length records in a file must be the same size, the variable-length portion can vary. This means that the overall length of each variable-length record in the file can vary.

When you create a Btrieve file that uses variable-length records, you specify the length in bytes of the fixed-length portion of each record. This length is the minimum length of each record. You do not define the maximum record length.

When you insert or update a record, the application uses the data buffer length parameter to specify the length of the record to Btrieve. If the data buffer length you specify is less than the defined fixed-length portion of the record, Btrieve returns a status code and does not insert or update the record.

If the data buffer length is equal to the length of the fixed-length portion of the record, Btrieve will store none of the variable-length portion of the data.

When you attempt to read a record longer than the length specified in your data buffer length parameter, Btrieve returns as much of the record as possible, based on the data buffer length you specified. Btrieve then returns Status Code 22 (The data buffer parameter is too short) to inform you that it did not read the entire record.

If you specify a data buffer length longer than the record you want to retrieve, Btrieve returns only the number of bytes in the actual record. In all cases, Btrieve informs the application of the number of bytes it returned by setting the data buffer length parameter to that value.

Storing Variable-Length Records: Variable Tails and VATs

In a Btrieve file, each page is the same size, and the maximum length of a page is 4,096 bytes. Btrieve splits the variable-length portion of a record between as many variable pages as necessary to accommodate the record's length.

Accordingly, for a file with a page size of 4,096 bytes and a variable-length portion that is 409,600 bytes long, Btrieve would split the record between at least 100 variable pages (409,600/4,096). Btrieve connects these pages (or more precisely, the parts of these pages devoted to the record) as a linked list. This linked list is called a *variable tail*.

If an application uses chunk operations to access part of a record, and that part begins at an offset well beyond the beginning of the record itself, Btrieve may spend considerable time reading the variable-tail linked list to seek to that offset.

To limit such seek time in nonsequential access to very large records, Btrieve 6.1 introduced a new file structure called the Variable-tail Allocation Table (VAT). Btrieve stores the VAT on variable pages. In a file containing VATs, each record that has a variable-length portion has its own VAT.

Using a record's VAT, Btrieve can divide the variable-length portion of that record into smaller portions and then store those portions in any number of variable tails. Btrieve stores an equal amount of the record's data in each of the record's variable tails (except the final variable tail).

Btrieve can use the VAT to accelerate a seek to a large offset in a record because the VAT allows it to skip reading the variable tails containing the record's lower-offset bytes.

Free Space List

Btrieve maintains the Free Space List for variable pages. The Free Space List indicates which variable pages contain the same or more free space than that specified by the Free Space Threshold file specification.

The Free Space Threshold file specification (discussed in “Free Space Threshold” on page 284 in Appendix B) is a value you can specify when you create a file. This value, expressed as a percentage, tells Btrieve how much free space must remain on a variable page in order for that page to appear on the Free Space List.

When Btrieve adds new variable-length records to the file, it uses pages in the Free Space List before using new variable pages. After each insert, update, or delete operation, Btrieve rechecks the remaining space on the affected variable page to see if it is still above the threshold to qualify for the Free Space List.

The free space threshold feature provides a means of reducing the fragmentation of variable-length records across several pages. A higher free space threshold reduces fragmentation at the cost of requiring more disk space for the file.

Compressed Records

If you specify the Data Compression option (discussed in “Data Compression” on page 263) when you create a Btrieve file, Btrieve compresses the file’s records before inserting or updating them and uncompresses the records when it retrieves them.

The length of a record after compression depends on its contents. Even records whose lengths are equal before compression are likely to have different lengths after compression. For this reason, Btrieve stores compressed records on variable pages, even if during the create operation the file was specified as *not* allowing variable-length records.

A compressed fixed-length-record file differs from a compressed variable-length-record file only in that Btrieve prevents insert and update operations from producing a record longer than a compressed fixed-length-record file’s specified record length.

Compressed record files do, however, have data pages. A record’s entry on its data page establishes an address for the record so that the Btrieve Step and Get

Direct/Record operations can find the record. (See “Accessing Records by Physical Location” on page 233.)

The data page entry can also provide links to other records sharing a duplicate key value, and it provides a pointer to the record’s contents, which are compressed and stored on variable pages.

Keys

Btrieve uses keys to allow fast direct access to records in a Btrieve file. Because Btrieve has no way of knowing the structure of the records in each file, you define each key by identifying the following:

- The key’s offset in bytes from the beginning of the record
- The number of bytes you want to use for the key
- The key’s type (identifying how sorting is to be performed)

For example, suppose a particular key begins at the eighth byte of the record and extends for four bytes. When you insert a record into the file, Btrieve extracts four bytes, beginning at the eighth byte, and uses this extracted value to position the record in the index.

When you create a key, you can assign attribute and type information to it. Key attributes and types are closely related and must be used in conjunction with each other.

Appendix B, “Description Files,” discusses the various key attributes and key types in term of specifying these items when you create a description file.

Key Attributes

The following key attributes are available:

- Segmented and Nonsegmented Keys
- Duplicatable and Nonduplicatable Keys
- Linked-Duplicatable and Repeating-Duplicatable Keys (6.*x*)
- Supplemental Keys (pre-6.*x*)
- Modifiable and Nonmodifiable Keys
- Descending and Ascending Keys (sort order)
- Case-Insensitive and Case-Sensitive Keys (sort order)
- Alternate Collating Sequence (sort order)
- Null Keys (All-Segment and Any-Segment)

Segmented and Nonsegmented Keys

Keys can consist of one or more *segments* in each record. A segment can be any set of contiguous bytes in the record. The total length of a key equals the sum of the length of the key segments, and the maximum length is 255 bytes. Different key segments may overlap each other in the record.

A Btrieve file limits the maximum number of key segments (rather than the maximum number of keys). The maximum number of key segments allowed depends on the page size, as shown in Table 9.

Table 9
Page Size and Maximum Number of Key Segments

Page Size	Maximum Key Segments
512	8
1,024	23
1,536	24

Table 9 *continued*

Page Size and Maximum Number of Key Segments

Page Size	Maximum Key Segments
2,048	54
2,560	54
3,072	54
3,584	54
4,096	119

A file with a page size of 512 bytes may contain 1 key with 8 segments, 8 keys with 1 segment each, or any combination in between. If a file has a page size of 1,024 bytes, it may contain 1 key with 23 segments, 23 keys with 1 segment, or any combination in between.

The key type can be different for each segment in the key. The sort order, either ascending or descending, can be different for each segment as well.

When a segmented key is a nonduplicatable key (see the following section), the combination of the segments must form a unique value. However, individual segments may contain duplicates. Duplicatable keys

When you are defining this type of segmented key, each segment would have `duplicates=no` as a key-level attribute even though that particular segment could have duplicates. To ensure that a particular segment is always unique, define it as a separate nonduplicatable key in addition to the segmented key definition.

Duplicatable and Nonduplicatable Keys

If you define a key to be duplicatable, Btrieve allows more than one record to have the same value for that key. If you specify a key as having no duplicates, Btrieve does not allow an application to add multiple records with the same value as this key.

For example, in a file containing customer records, you can define the zip code as a duplicatable key so that many different records can contain the same value for the zip code. However, if you also make the customer number a key, you probably would not want to allow duplicates, since each customer should have a unique number.

Note If one segment of a segmented key allows duplicates, all the segments must allow duplicates. For more information, "Segmented and Nonsegmented Keys" on page 212.

For information about how Btrieve stores duplicate key values, see the following section and "Indexes" on page 231.

Linked-Duplicatable and Repeating-Duplicatable Keys

By default in a 6.*x* file, Btrieve stores duplicatable keys as *linked*-duplicatable keys.

In this situation, Btrieve stores the key extracted from the first record of a duplicatable key on an index page. When the first record containing a duplicate value for a key is inserted into a file, Btrieve does not store the duplicate key value on the index page.

Instead, it goes to the original record (the one whose key value is duplicated in the new record) and stores a *pointer* at the end of the record. The pointer points to the new record.

As each new record with a duplicate key value is inserted into the file, Btrieve stores a pointer to that record in the preceding record containing the same key value. In addition to having pointers that point to the next record containing a duplicate key value, Btrieve has pointers that point to the preceding record containing that duplicate key value. Btrieve uses the pointers in its Get Next and Get Previous operations.

This method of storing pointers at the end of data records (as opposed to storing keys and values on index pages) forms a doubly linked list of records with keys containing duplicate values, resulting in the name *linked*-duplicatable keys.

Note If you create a file with linked-duplicatable keys, Btrieve reserves space in each record of the file for that key’s duplicate pointer.

If no room is available to create a linked-duplicatable key (that is, if no duplicate pointers were reserved at file creation, or if no index has been dropped to free existing pointers), Btrieve creates a *repeating*-duplicatable key.

Btrieve stores every key value of a repeating-duplicatable key both on a data page and on an index page. In other words, the key’s value resides in the record on a data page and is *repeated* in the key entry on an index page.

By default in 6.*x* files, Btrieve stores duplicatable keys as linked-duplicatable keys. However, Btrieve 6.1*x* allows you to override that default.

Modifiable and Nonmodifiable Keys

You can also define a key as modifiable or nonmodifiable. If you define a key as modifiable, Btrieve allows your application to update an existing record and change the value of the key.

For example, if the account balance is a key, you may allow a program to modify the value of that key as the customer makes purchases and payments. However, if the customer’s account number is a key, you probably would make it a nonmodifiable key because the customer’s account number should not change.

Note If one segment of a key is modifiable, all the segments must be modifiable.

Descending and Ascending Keys

Btrieve normally orders key values in ascending order (lowest to highest). However, you can specify that Btrieve order the key values in descending order (highest to lowest) when you define the key segment.

When an application performs a Get Greater (8) operation on a descending key, Btrieve returns the record corresponding to the first key value that is lower than the key value you specify in the key buffer.

For example, consider a file which has 10 records and a descending key of type `integer`. The actual values stored in the 10 records for the descending key are the integers 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. If the current record’s key value is 5 and you perform a Get Greater operation, Btrieve returns the record containing the key value 4.

Similarly, when an application performs a Get Less Than (10) operation using a descending key, Btrieve returns the record with the next higher value than the one you specify in the key buffer.

Using the preceding example, if the current record’s descending key has a value of 5 and the application performs a Get Less Than operation, Btrieve returns the record containing the key value 6.

Case-Insensitive and Case-Sensitive Keys

By default, Btrieve is case sensitive when sorting string keys—that is, it sorts string key values based on whether letters are uppercase or lowercase. Btrieve sorts the values by putting the uppercase value first in the sort. For example, *AAA* and *BBB* would sort before *aaa*.

To make Btrieve ignore case when sorting string keys, you can assign the case-insensitive attribute to a key. When you define a key to be case insensitive, Btrieve assigns the same collating value to characters *a* through *z* when sorting the key as it assigns to the corresponding characters *A* through *Z*.

Note Case sensitivity does not apply if Btrieve sorts a key according to the collating weights in an ACS.

Alternate Collating Sequence

You can also direct Btrieve to sort string keys (types `lstring`, `zstring`, and `string`) differently from the standard ASCII collating sequence.

By using one or more alternate collating sequences (ACSs), you can sort keys in one of the following ways:

- By a locale-specific collating sequence (such as the German, Swedish, or Finnish character sets)
- By your own user-defined sorting order, which may require a sorting sequence that mixes alphanumeric characters (A–Z, a–z, and 0–9) with nonalphanumeric characters (!, #, and so on).

Note

Only files using the format for Btrieve 6.1*x* or later can have more than one ACS.

Locale-Specific ACS

Btrieve 6.1*x* allows you to sort an index by languages other than English. This locale (national language) support originates in the operating system. Therefore, refer to your operating system's documentation for specific information.

When you create a key of type `string`, `lstring`, or `zstring` (described in "Key Types" on page 224), you can instruct Btrieve to sort the values for that key according to a specified locale's collating sequence. The application can designate the locale by passing to Btrieve a locale's country ID and code page number.

Alternatively, the application can specify the ACS for the default locale—that is, the locale for which the operating system running Btrieve is configured. For client applications making calls to server-based Btrieve, this is the server's locale.

When an application calls Btrieve to create an index that is to be sorted according to a locale-specific ACS, Btrieve queries the operating system to retrieve the desired collating sequence. When the operating system returns the collating sequence, Btrieve stores it in the Btrieve file.

If the Btrieve file is moved to a different locale, or if the same computer is reconfigured to support a different locale, Btrieve still sorts the index according to the original locale's collating sequence, thereby allowing Btrieve to insert new key values correctly.

Note

Because Btrieve obtains the locale's collating sequence by making operating system calls, it cannot successfully create an index for a locale that is not supported by the operating system. Instead, Btrieve returns an error code to the application. The NetWare operating system supports only one locale at a time, so only the server's default locale is supported.

To create an ACS that will sort a Btrieve file according to a locale-specific collating sequence, an application must place 265 bytes directly into the data buffer of a Create (14) operation, using the format shown in Table 10.

Table 10
Locale-Specific Alternate Collating Sequence Format

Offset	Length	Description
0	1	Signature byte. This byte should contain the value AD hex.
1	2	Country ID (Intel* format). See your operating system's documentation for more information on national language support.
3	9	Code page ID (Intel format). See your operating system's documentation for more information on national language support.
5	260	Filler.

The 265-byte ACS definition should follow the last key specification block in the data buffer.

Note

To use the default locale-specific ACS, specify a value of 0xFFFF for the Country ID and the Code page ID.

User-Defined ACS

To create an ACS that will sort a Btrieve file according to a user-defined collating sequence, the application must place 265 bytes (using the format shown in Table 11) directly into the data buffer of a Create (14) operation following the last key specification block in the data buffer.

Table 11
User-Defined Alternate Collating Sequence Format

Offset	Length	Description
0	1	Signature byte. This byte should contain AC hex.
1	8	An 8-byte name that uniquely identifies the ACS to Btrieve.

Table 11 *continued*

User-Defined Alternate Collating Sequence Format

Offset	Length	Description
9	256	A 256-byte map. Each 1-byte position in the map corresponds to the code point having the same value as the position’s offset in the map. The value of the byte at that position is the collating weight assigned to the code point. For example, to force code point 0x61 (’a’) to sort with the same weight as code point 0x41 (’A’), place the same values at offsets 0x61 and 0x41.

Following are a 9-byte header and a 256-byte body that represent a collating sequence named UPPER. The header appears as follows:

```
AC 55 50 50 45 52 20 20 20
```

The 256-byte body appears as follows (with the exception of the offset values in the leftmost column):

```
00: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10: 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20: 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
30: 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40: 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50: 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60: 60 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
70: 50 51 52 53 54 55 56 57 58 59 5A 7B 7C 7D 7E 7F
80: 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90: 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0: B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0: C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0: D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0: E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0: F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
```

The header and body forming this ACS are shipped with this version of Btrieve as the file UPPER.ALT. UPPER.ALT provides a way to sort keys without regard to case, although Btrieve 6.*x* makes this method obsolete by allowing an application to specify in bit 10 of a key’s flag values whether the key is to be sorted using case sensitivity.

However, UPPER still provides a good example for use when writing your own ACS.

Offsets 0x61 through 0x7A in the example have been altered from the standard ASCII collating sequence. In the standard ASCII collating sequence, offset 0x61 contains a value of 0x61 (representing lowercase ’a’). When a key is sorted with the UPPER ACS, Btrieve sorts lowercase ’a’ (0x61) with the collation weight found at offset 0x61: 0x41. Thus, the lowercase ’a’ is sorted as if it were uppercase ’A’ (0x41).

Therefore, for sorting purposes UPPER converts all lowercase letters to their uppercase equivalents when sorting a key.

Each 1-byte position in the map corresponds to the code point having the same value as the position’s offset in the map. The value of the byte at that position is the collating weight assigned to the code point.

For example, to force code point 0x61 (’a’) to sort with the same weight as code point 0x41 (’A’), place the same values at offsets 0x61 and 0x41.

The following 256-byte body basically performs the same function as UPPER.ALT’s body except that ASCII characters preceding the ASCII space (0x20) are now sorted *after* all other ASCII characters:

```
00: E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
10: F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
20: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
30: 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
40: 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
50: 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
60: 40 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
70: 30 31 32 33 34 35 36 37 38 39 3A 5B 5C 5D 5E 5F
80: 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
90: 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
A0: 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
B0: 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
C0: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
D0: B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
E0: C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
F0: D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
```

In this body, different collating weights have been assigned so that a character’s weight no longer equals its ASCII value. For example, offset 0x20, representing the ASCII space character, has a collating weight of 0x00; offset 0x41, representing the ASCII uppercase ‘A’, has a collating weight of 0x21.

To achieve the ability to sort keys without regard to case, offsets 0x61 through 0x7A in the last example have been altered. As in the body for UPPER.ALT, offset 0x61 now has the same collating weight as offset 0x41: 0x21. By having the same collating weight, offset 0x41 (uppercase ‘A’) sorts the same as offset 0x61 (lowercase ‘a’).

ACSs and Multiple Btrieve Files

If you have multiple files with different ACSs, each sequence should have a different name to avoid confusing two sequences that have the same name but that collate differently.

Multiple ACSs in a Single Btrieve File

You can use multiple ACSs in a single Btrieve file. Btrieve allows one ACS per key. Therefore, if the key is segmented, each segment must use either the key’s specified ACS or no ACS at all.

In a Btrieve file in which a key has an ACS designated for some segments but not for others, Btrieve sorts only the ACS segments by the specified ACS.

Null Keys (All-Segment and Any-Segment)

Note Previous editions of the Btrieve documentation used the term *manual key*. This edition replaces that term with *any-segment null key*.

You can direct Btrieve to exclude certain records from an index by defining null keys. When you define a null key, you specify a null value by which Btrieve can recognize the key as being a null key.

You can use a null value in a key when the data for the key is unavailable or when you do not want Btrieve to include a record for that key in the index. Null keys allow you to avoid searching through meaningless records in an index path and to eliminate the overhead time required to update the index each time a key with no meaningful data is inserted.

When data becomes available for that key, and when you want to include the record in the index, you can update the record, replacing the null value with meaningful data.

If you define one segment of a key as a null key, you must define all segments of that key as a null key. However, Btrieve allows you to define different null values for different segments in a segmented key. The most commonly used null values are ASCII blank (0x20) and binary 0 (0x00).

Btrieve can use one of two different methods to determine whether a key is a null key. You determine which method Btrieve uses by specifying one of the two possible null key attributes:

- All segments
- Any segment

If you specify the all-segment null key attribute, Btrieve excludes from the index those records in which *all* the bytes in *all segments* of a key contain a null value. (Previous editions of the Btrieve documentation called this attribute the *null key attribute*.)

If you specify the any-segment null key attribute, Btrieve excludes from the index those records in which every byte in *any* segment of a key contains the null value. (Previous editions of the Btrieve documentation called this attribute the manual key attribute.)

Often, an any-segment null key is used as a flag that indicates whether the key is to be indexed. If the segment contains only the null value that you defined for that segment, Btrieve does not include the key in the index even though the rest of the segments in the key contain non-null values.

By updating the flag segment with any value other than the specified null value, you can instruct Btrieve to include the record in the index.

Important

If you specify both the all-segment and any-segment null key attributes, Btrieve uses only the any-segment null key attribute.

The following diagram shows three segments of a key in a given record:

20	20	20	.	.	.

00	00	00	.	.	.

20	20	20	.	.	.

The null value defined for the first and last segments of the key is the ASCII blank (0x20). The null value defined for the middle segment is binary 0 (0x00). As the diagram shows, all the bytes in each segment contain their defined null value. Therefore, Btrieve excludes this record from its index if you specify *either* the all-segment or the any-segment null key attribute.

The following diagram shows three segments of a key in a different record:

20	20	20	.	.	.

42	54	52	49	45	56

45	52	55	4C	45	53

The null values defined for the segments in this diagram are the same as before (0x20 for the first and last segment, 0x00 for the middle segment).

As the second diagram shows, all the bytes in the first segment contain their defined null value. However, the second and third segments contain non-null values. Therefore, Btrieve excludes this record from its index only if you specify the any-segment null key attribute. If you specify the all-segment null key attribute, Btrieve includes the record in its index.

The second diagram shows a situation in which the first key segment could be used as a flag to indicate whether the key is to be indexed.

For example, an application would first define the segments as any-segment null keys. If the application wanted to include a record in the index, it would place a non-null value in the first segment. Conversely, if the application did not want to include a record in the index, it would place a null value in the first segment.

Key Types

Btrieve provides standard and extended key types. These types allow Btrieve to recognize and collate key values based on the internal storage format of the 15 data types that compilers most commonly use. This capability provides greater flexibility when you design the keys of your Btrieve files.

For historical reasons, the two standard key types, `string` and `unsigned binary`, are also offered as extended key types.

Internally, Btrieve compares string keys on a byte-by-byte basis, from left to right. Btrieve sorts string keys according to their ASCII value. However, for string keys, you can specify either case-insensitive sorting or an ACS.

Btrieve compares unsigned binary keys a word at a time. It compares these keys from right to left because the Intel 8086 family of processors reverses the high and low bytes in an integer.

Table 12 lists the extended key types and their associated codes.

Table 12
Extended Key Types and Codes

Type	Code
string	0
integer	1
float	2
date	3
time	4
decimal	5
money	6
logical	7
numeric	8
bfloat	9
lstring	10
zstring	11
unsigned binary	14
autoincrement	15
sign trailing separate	17

The following sections, arranged alphabetically by key type, describe the extended key types.

autoincrement

An `autoincrement` key is a signed Intel-style integer that can be either two or four bytes long. Internally, `autoincrement` keys are stored in Intel binary integer format, with the high-order and low-order bytes reversed within a word.

Btrieve sorts `autoincrement` keys by their absolute values, comparing the values stored in different records a word at a time from right to left. `Autoincrement` keys are incremented each time you insert a record into the file.

The following restrictions apply to `autoincrement` keys:

- An `autoincrement` key cannot allow duplicate key values.
- An `autoincrement` key cannot be segmented or included as a segment of another key.
- An `autoincrement` key cannot overlap another key.

Btrieve treats `autoincrement` key values as follows when you insert records into a file:

- *If you specify a value of binary 0 for the* `autoincrement` *key*, Btrieve assigns a value to the key based on the following criteria:
  - If you are inserting the first record in the file, Btrieve assigns the value of 1 to the `autoincrement` key.
  - If records already exist in the file, Btrieve assigns the key a value that is one number higher than the highest existing absolute value in the file.
- *If you specify a nonzero value for the* `autoincrement` *key*, Btrieve inserts the record into the file and uses the specified value as the key value. If a record containing that value already exists in the file, Btrieve returns a nonzero status and does not insert the record.

When you delete a record containing an `autoincrement` key, Btrieve completely removes the record from the file. Btrieve does not reuse the deleted key value unless you specify that value when you insert another record into the file.

As mentioned previously, Btrieve always sorts `autoincrement` keys by their absolute values. For example, you can handle this key type in the following ways:

- Specify a negative value for an `autoincrement` key when you insert a record.
- Update a record and negate the value for the `autoincrement` key.

Either way, Btrieve sorts the key according to its absolute value. This allows you to use negation to flag records without altering the record's position in the index. In addition, when an application performs a Get operation and you specify a negative value in the key buffer, Btrieve treats the negative value as the absolute value of the key.

bfloat

A field with a `bfloat` type is a single- or double-precision real number. A single-precision real number is stored with a 23-bit mantissa, an 8-bit exponent biased by 128, and a sign bit.

The representation of a double-precision real number is the same as that for a single-precision real number except that the mantissa is 55 bits instead of 23 bits. The least significant 32 bits are stored in bytes 0 through 3. The `bfloat` type is commonly used in BASIC applications.

date

A `date`-type field is stored internally as a 4-byte value. The day and the month are each stored in 1-byte binary format. The year is a 2-byte binary number that represents the entire year value.

Btrieve places the day into the first byte, the month into the second byte, and the year into a two-byte word following the month.

decimal

A `decimal`-type field is stored internally as a packed decimal number with two decimal digits per byte.

The sign nibble is either 0xF or 0xC for positive numbers and 0xD for negative numbers. The decimal point is implied—that is, no decimal point is stored in the `decimal`-type field. The application is responsible for tracking the location of the decimal point for the value in a `decimal`-type field.

All the values for a `decimal` key type must have the same number of decimal places in order for Btrieve to collate the key correctly. The `decimal` type is commonly used in COBOL applications.

float

A `float` type is consistent with the IEEE standard for single- and double-precision real numbers. The internal format for a 4-byte `float` consists of a 23-bit mantissa, an 8-bit exponent biased by 127, and a sign bit.

A `float`-type field with 8 bytes has a 52-bit mantissa, an 11-bit exponent biased by 1,023, and a sign bit.

integer

An `integer` type is a signed whole number and must contain an even number of bytes. Internally, `integer`-type fields are stored in Intel binary integer format, with the high-order and low-order bytes reversed within a word.

Btrieve evaluates the key from right to left, a word at a time. The sign must be stored in the high bit of the rightmost byte. The `integer` type is commonly used in C language applications.

logical

The `logical` extended key type is stored as a 1- or 2-byte value. Btrieve collates `logical` key types as a string. This allows an application to determine the stored values that represent true or false.

lstring

An `lstring` type in Btrieve has the same characteristics as a regular `string` type except that the first byte of the string contains the binary representation of the string's length. The length stored in byte 0 of the `lstring` determines the number of significant bytes. Btrieve ignores any values beyond the specified length of the string.

The `lstring` type is commonly used in Pascal applications.

money

The internal representation of the `money` type is the same as that for the `decimal` type.

numeric

`Numeric` values are stored as ASCII strings, right justified with leading zeros. Each digit occupies one byte internally. For positive numbers, the rightmost digit can be represented by 1 through 0 instead of A through {. Btrieve processes positive numbers represented either way.

The `numeric` type is commonly used in COBOL applications.

sign trailing separate

The `sign trailing separate` data type (sometimes called `numeric STS`) is a COBOL data type that has values resembling those of the `numeric` data type. `Sign trailing separate` values are stored as ASCII strings and right justified with leading zeros.

However, the rightmost byte of a `sign trailing separate` string is either '+' (ASCII 0x2B) or '-' (ASCII 0x2D). This differs from `numeric` values that embed the sign in the rightmost byte along with the value of that byte.

string

A `string` type in Btrieve is a sequence of characters ordered from left to right. Each character is represented in ASCII format in a single byte except when Btrieve is determining whether a key value is null. (See "Null Keys (All-Segment and Any-Segment)" on page 222.)

time

A `time`-type field is stored internally as a 4-byte value. Hundredths of a second, minute, and hour are each stored in 1-byte binary format. Btrieve places the hundredths of a second value into the first byte, followed respectively by the second, minute, and hour values.

unsigned binary

Btrieve sorts `unsigned binary` keys as unsigned `integer` keys. An `unsigned binary` key must contain an even number of bytes (2, 4, 6, and so on). Btrieve compares `unsigned binary` keys a word at a time, from right to left.

An `unsigned binary` key is sorted in the same manner as an `integer` key. The differences between an `unsigned binary` key and an `integer` key are that an `integer` has a sign bit, while an `unsigned binary` type does not, and an `unsigned binary` key can be longer than 4 bytes.

zstring

A `zstring` type in Btrieve corresponds to a C `string`. It has the same characteristics as a regular `string` type except that a `zstring` type is terminated by a byte containing a binary 0. Btrieve ignores any values beyond the first binary 0 it encounters in the `zstring` except when Btrieve is determining whether a key value is null. (See “Null Keys (All-Segment and Any-Segment)” on page 222.)

Indexes

An index is a structure in a Btrieve file that contains the key values for a specific key. In an index, Btrieve dynamically maintains the key values in a sorted order, storing them in a balanced B-tree structure. When you insert, update, or delete a record, Btrieve adjusts all the indexes for the file to reflect the latest changes in the key values contained in the records.

Btrieve keeps all indexes to the data records in the form of *standard B-trees.* A B-tree is a data structure that allows external searching by means of multiway tree branching. The B-tree structure features quick access and efficient use of disk space. Once a B-tree structure is created, no periodic maintenance is required. Btrieve creates a separate B-tree for each key you define within a file.

You can create indexes when you are creating the Btrieve data file, or at any time thereafter. When you create a Btrieve data file, you can define one or more keys for Btrieve to use in building indexes.

You can also define external indexes (key-only files, described on page 204) for a Btrieve file after creating the file. Positioning rules (guidelines governing which record is current, which is next, and so on) are the same, regardless of when you create an index.

The total number of indexes (actually, the total number of key segments) for a file depends on the file's page size:

Page Size	Maximum Key Segments
512	8
1,024	23
1,536	24
2,048	54
2,560	54
3,072	54
3,584	54
4,096	119

You can delete, or *drop*, an index when your application no longer needs it. The space that the index used in the file is freed for data or for other index pages.

You can also use index balancing to conserve disk space (see “Index Balancing” on page 264).

If you create an index at file creation time, Btrieve stores duplicate values in the chronological order in which the records are inserted into the file.

Important

The chronological ordering of records can change when you update records and change their key values, when you drop and rebuild an index, or when you rebuild the Btrieve file. An application should not depend on the chronological ordering of records in a Btrieve file.

If you create a new index for a file that was created earlier, Btrieve enters duplicate values into the index in the order in which their corresponding records are physically stored in the file at the time the index is created. For linked-duplicatable keys, additional duplicate values are added to the end of the series of duplicates as new records are inserted.

How Btrieve stores duplicate key values in an index also depends on whether the key is a linked-duplicatable or a repeating-duplicatable key. For more information, see “Linked-Duplicatable and Repeating-Duplicatable Keys” on page 214.

You can also specify a key with an alternate collating sequence. For more information, see “Alternate Collating Sequence” on page 216.

Accessing Records

Btrieve allows you to access records from a file based on either the record's physical address within the file or a key value contained in the record. Btrieve also allows you to access portions of a record by using chunk operations.

Accessing Records by Physical Location

Record accessing by physical location is faster for the following reasons:

- Btrieve does not have to use any index pages.
- The next or previous physical record is usually already in Btrieve's memory cache because the page on which it resides is probably in cache.

The following sections explain physical currency (the effect on positioning when accessing records by physical location) and the Step operations (the record operations used to access a record by its physical location).

Physical Currency

When you insert a record into a Btrieve file, Btrieve writes the record into the first free space available in the file, regardless of any key values contained in the record. This location is referred to as the *physical location*, or address, of the record. The record remains in this location until you delete it from the file.

The physical location of the records in the file determines the *physical order* of the records. The Btrieve Step operations use the physical location to access records.

The physical current, next, and previous locations together form the *physical currency* within the file.

Step Operations

An application can use the Step operations to access records based on their physical location within a file.

For example, the Step First (33) operation retrieves the record that is stored in the first, or lowest, physical location in the file. The Step Next (24) operation retrieves the record stored in the next higher physical location.

The Step Previous operation retrieves the record stored in the next lower physical location in the file. The Step Last (34) operation retrieves the record that is stored in the last, or highest, physical location in the file.

The Step Next Extended (38) and Step Previous Extended (39) operations retrieve one or more records from the physical location following or preceding the current record.

The Step operations are useful for traversing a Btrieve file quickly if an application does not need to retrieve the records in a specific order.

Accessing Records by Key Value

Accessing records by key value allows you to retrieve records based on their values for a specified key. The following sections discuss these topics:

- Logical currency—The effect on positioning when accessing records by key value
- Get operations—The Btrieve operations used to access a record by its key value
- The position block—The location used to store logical positioning information

Logical Currency

When you insert a record into a file, Btrieve updates each B-tree index for which the appropriate key in the record has a non-null value. Each key of a file determines a logical ordering of the records. The ordering can be from low to high or from high to low, depending on whether the key is defined as ascending or descending.

The record accessed last is not only the physical current record, but also the logical current record. For example, when you insert a record, that record becomes the physical current record. It also becomes the logical current record (unless you perform a no-currency change Insert operation or the inserted record's key value is null).

Relative to any logical current record based on a certain key, the record next in the defined logical sequence is the logical next record and the record previous in the defined logical sequence is the logical previous record.

Note The logical previous of the logical first record does not exist—nor does the logical next of the logical last record.

The logical current, next, and previous locations together form the logical currency within the file.

The Btrieve Get record operations use the logical locations when accessing records.

Get Operations

An application can use the Get operations to retrieve records based on their values for a specified key. The appropriate Get operation can retrieve a specific record from a file or retrieve records in a certain order.

For example, the Get First (12) operation retrieves the first record by the key specified in the key number parameter. Likewise, the Get Last (13) operation retrieves the last record according to the logical order based on the specified key.

Some Get operations, such as Get Equal (5) or Get Less Than (10), return a record based on a key value your application specifies in the key buffer parameter.

Get operations establish Btrieve's logical currency in an index. In addition to establishing logical currency, all Get operations except Get Position also establish the physical currency.

Position Block

Btrieve uses an area of memory called the *position block* to maintain logical positioning information for use in accessing records. Btrieve maintains positioning and other necessary information associated with each open file. It stores this information in a 128-byte block of memory that passes between Btrieve and your application.

Some of the information contained in the position block is as follows:

- The current index path identifier (the key number).
- Three record pointers reflecting the logical currency based on the current key number:
  - The *previous* record pointer points to the logical previous position.
  - The *current* record pointer points to the most recently *accessed* record. This record is not necessarily the most recently retrieved record.
  - The *next* record pointer points to the logical next position.

Many of the Btrieve operations modify the contents of the position block to reflect the new position within the file. Btrieve uses positioning information to read sequentially through the file by a given key.

An application should maintain a 128-byte position block for every Btrieve file that is open.

Accessing Records by Chunks

A chunk is a set of logically contiguous bytes that is defined by its offset and length within a record and that can be accessed by the Btrieve Get Direct/ Chunk (23) operation or by an Update Chunk (53) operation. This ability to operate on a chunk (rather than on the entire record) is a feature introduced in Btrieve 6.1 and is applicable to any 6.*x* file.

Note Chunks are defined only for the duration of the operation that defines them.

A chunk's offset and length need not correspond to any of the internal structures of a record that are known to Btrieve, such as key segments, the fixed-length portion of a record, or a variable tail (discussed on page 209).

Also, a chunk need not correspond to any portion of the record defined by the application (for example, a field), although you may find it useful to update such defined portions as chunks.

Probably the main reason to use chunks in an application is to overcome the restrictions imposed by the Btrieve data buffer length parameter.

Essentially, an application can pass no more than 65,535 bytes in the Btrieve data buffer at one time (sometimes less). Therefore, if the records in an application exceed this length, you cannot pass an entire record in a single Btrieve operation. The application can, however, divide that record into chunks.

An application using chunk operations also allows a workstation that makes calls to a remote Btrieve server to use a smaller value for its requester's Maximum Data Buffer Length parameter. This smaller value lowers the requester's requirement for memory on the workstation.

Btrieve provides two operations that allow an application to access chunks: Get Direct/Chunk (23) and Update Chunk (53).

Using Btrieve Transactions

If you have a number of modifications to make to a file, and you must be sure that either *all* or *none* of those modifications are made, include the operations for making those modifications in a Btrieve transaction.

To include a group of operations within a Btrieve transaction, you enclose those operations between a Begin Transaction (19) operation and an End Transaction (20) operation.

Btrieve commits to disk the changes made inside a transaction only when the application performs the End Transaction operation. Prior to performing the End Transaction operation, Btrieve makes all modifications in memory. (See “Shadow Paging” on page 245.)

Should a system failure occur before the application finishes performing all the operations included in the transaction, Btrieve ignores any changes previously made within the transaction.

When you restart the system after the failure, the Btrieve file contains the same information it contained before the Begin Transaction operation was performed.

NetWare Btrieve 6.1*x* provides two types of transactions:

- Exclusive
- Concurrent

The type you use depends on how severely you want to restrict other clients' access to the file you are modifying.

Prior to 6.0, Btrieve allowed only exclusive transactions. When a Btrieve task operates on a file inside an exclusive transaction, Btrieve locks that *entire file* for the duration of the transaction. Once a file is locked in an exclusive transaction, other clients can read the file, but they cannot make changes to it.

As this inability to make changes implies, a Btrieve client (client 2, for example) can begin an exclusive transaction on a file, even if another client (client 1, for example) currently has the file (or a portion of it) already locked. However, client 2 cannot perform any operation that requires a position block (even standard Get or Step operations) until client 1 releases whatever locks it has on the file.

Versions of Btrieve prior to 6.0 allowed other applications (or, in the case of NetWare Communication Services, other *clients*) to see the changes to the files involved in an exclusive transaction before the transaction ended.

Btrieve versions 6.0 and later do not allow other applications or clients to see the changes involved in *any* transaction (exclusive or concurrent) until the transaction ends, regardless of which version of Btrieve was used to create the file originally.

When an application operates on a file inside a concurrent transaction, Btrieve locks only the affected records, or (at most) only the affected pages in the file.

Btrieve locks the data page that contains a record being modified in an insert, update, or delete operation. Additionally, if the record is a variable-length record, Btrieve locks all the variable pages containing portions of the record. Finally, Btrieve locks any index pages that will be modified as a result of the insert, update, or delete operation.

As with exclusive transactions, other tasks can always read data that is locked from within a concurrent transaction.

In any Btrieve data file, multiple tasks can have their own concurrent transactions operating, in which they are performing insert, update, or delete operations, or in which they are performing Get or Step operations that contain read lock biases. The only restriction on these activities is that no two tasks can lock the same record or page simultaneously from their respective concurrent transactions.

The following additional features apply to concurrent transactions:

- The file must use the 6.*x* format. (Otherwise, operation 1019 will be treated as operation 19: an exclusive transaction.)
- Locked pages remain locked for the duration of the transaction.
- If the transaction simply reads a record, Btrieve does *not* lock either the record or the corresponding page.
- Other clients *cannot* see the changes to a file in a concurrent transaction until the transaction ends.

Supporting Multiple Btrieve Clients

Btrieve provides several concurrency control methods and tools to resolve conflicts that can occur when multiple Btrieve clients attempt to access or modify records in the same file concurrently:

- Passive Concurrency
- Locks
  - Explicit Record Locks
  - Implicit Record Locks
  - Implicit Page Locks
  - File Locks

The following sections discuss these methods in detail.

Passive Concurrency

You may choose to rely on passive concurrency for resolving update conflicts if an application will perform single-record read and update operations in the following situations:

- While not inside a transaction
- From within a concurrent transaction

With passive concurrency, Btrieve allows a client to read a record without applying any lock bias for the operation. Then, if a second client changes the record between the time the first client reads it and the time the first client attempts to update or delete it, Btrieve returns Status Code 80 (Conflict).

In this situation, the modification initiated by the first client would be based on an outdated image of the record. Therefore, the first client must read the record again before performing the Update (3) or Delete (4) operation.

The following tables show how two Btrieve clients interact when using passive concurrency. Table 13 illustrates non-transactional concurrency.

Table 13
Passive Concurrency (Non-transactional Example)

Application 1	Application 2
1. Open file.	
	2. Open file.
3. Read Record A.	
	4. Read Record A.
5. Update Record A.	
	6. Update Record A. Btrieve returns Status Code 80 (Conflict).
	7. Reread Record A.
	8. Update Record A.

Table 14 illustrates passive concurrency from within a concurrent transaction.

Table 14
Passive Concurrency (Concurrent Transaction Example)

Application 1	Application 2
1. Begin concurrent transaction.	
	2. Begin concurrent transaction.
3. Read record A.	
4. Update record A.	
	5. Read record A.
6. End transaction.	
	7. Update record A. Btrieve returns conflict status code.
	8. Reread record A.

Table 14 *continued*
Passive Concurrency (Concurrent Transaction Example)

Application 1	Application 2
	9. Update record A.
	10. End transaction.

Note Even though Application 2 reads record C *after* Application 1 has already executed the update operation, Btrieve correctly detects a conflict error in Step 7. This conflict exists because Application 1 does not commit the change it made to record A until ending its transaction in Step 6. By the time Application 2 attempts its update in Step 7, the image it read of record A (in Step 5) is outdated.

Locks

Records, pages, or even an entire Btrieve file may be *locked*. Once locked, a record, page, or file cannot be modified by any client other than the one responsible for the lock. Similarly, locks owned by one client can prevent record, page, or file locking by another client, as explained in the rest of this section.

Btrieve provides two kinds of locks:

- Explicit
- Implicit

When a client specifically requests the lock by including the lock request with a Btrieve operation code, that lock is called an *explicit* lock.

However, even when a client does not explicitly request a lock, Btrieve may lock an affected record or page as the result of some action performed by the client. In this situation, the lock requested by Btrieve is called an *implicit* lock.

- Records can be locked either explicitly or implicitly.
- Pages can only be locked implicitly.
- Files can be locked only explicitly.

The rest of this section discusses the various locks as they apply in both non-transactional and transactional environments.

Implicit Record Locks

When a client attempts to update or delete a record, either *external* to any transaction or from within a *concurrent* transaction, Btrieve *implicitly* tries to lock that record on behalf of the client (hence the term implicit record lock).

In an exclusive transaction, an implicit record lock is unnecessary because Btrieve locks the entire file prior to performing the update or delete operation.

Btrieve can grant an implicit record lock to a client as long as no other client

- Holds an explicit lock on the record
- Holds an implicit lock on the record
- Has locked the file containing the record

Implicit Page Locks

Btrieve tries, on behalf of the client, to implicitly lock the *pages* that are modified during execution of an insert, update, or delete operation if the modification occurs either outside of a transaction or from within a concurrent transaction.

Note

In an exclusive transaction, an implicit page lock is unnecessary because Btrieve locks the entire file prior to performing an update or delete operation. In the case of an Insert operation, Btrieve requests a file lock if the client does not have one yet.

As with implicit record locks, implicit page locks are provided by Btrieve. The client does not explicitly request them.

The data page containing the record being modified (or that will contain the record being inserted) must always be locked. However, a single operation might need to lock several other pages as well.

For example, if the change made to a record involves one or more of the record’s keys, then Btrieve must lock the index pages containing the affected key values. Btrieve must also lock all index pages modified by the action of balancing the B-tree(s) during the operation.

If a modification affects the variable-length portion of a record, Btrieve must lock the variable page(s) as well.

Implicit page locks allow multiple clients to modify different parts of the same file at the same time from within concurrent transactions, as long as those modifications do not affect other previously locked portions of the file.

File Locks

When a client “touches” a file for the first time in an exclusive transaction by performing an insert, update, or delete operation, that client tries to obtain a file lock.

Note As the preceding statement implies, Btrieve does *not* lock a file when the client performs a Begin Transaction operation. The lock occurs only when the client reads or modifies a record after performing the Begin Transaction operation.

A file lock, as its name suggests, locks the entire Btrieve file. A client’s file locks remain in effect until that client ends or aborts the transaction, or until the client is reset (which implies performing an Abort Transaction operation).

Recovering Data

Btrieve provides a number of methods for recovering data. Some of these methods are automatic and require no interaction from an application or an end-user. Others require some form of manual intervention.

Important None of the following methods eliminate the need to back up files because none of these methods can recover data from a damaged disk. Therefore, it is imperative that backups be made to protect against the catastrophic loss of a file because of a hardware failure.

Automatic Data Recovery

Btrieve provides the following automatic data recovery methods:

- Shadow paging or pre-imaging (depending on the file's format)
- Transaction control

Shadow Paging

For any Btrieve file using a 6.*x* format, Btrieve automatically guards against file corruption by using *shadow paging*.

Shadow paging is especially useful when Btrieve needs to perform several physical page updates to process either a single non-transactional request or an entire transaction for an application.

Without shadow paging, if Btrieve processing is interrupted during these operations by a system failure, files may be corrupted. Corrupted files may contain inconsistent index pages, and retrieving data from the file might be impossible. Shadow paging provides automatic protection against such corruption.

During the execution of a request or transaction that changes the file, Btrieve does not overwrite original pages. Instead, it creates *shadow pages* (copies of the original pages to be modified), then selects and locks free physical locations within the file where the pages will ultimately be stored. Btrieve then makes the requested modifications to the shadow pages where they reside in cache.

The original pages remain unchanged. When all modifications to a particular shadow page are finished, Btrieve initiates the process of writing that shadow page to a selected free physical location within the file.

For the purposes of data recovery, Btrieve bundles one or more non-transactional operations and/or committed transactions into a *system transaction*. A system transaction can include operations or transactions from one or more users. However, unless a system failure occurs, system transactions are completely transparent to an application.

If all changes to shadow pages are successful for a single non-transactional request, or if an End Transaction (20) operation is issued for a transaction, Btrieve commits the changes by making the shadow pages the new original pages (and therefore no longer shadow pages).

However, if partial changes made by a single request need to be rolled back, or if an Abort Transaction (21) operation is issued, Btrieve returns the space occupied by any shadow pages that are uncommitted (but that have already been written to disk) to one of Btrieve's *Free Space Lists*. Once on a Free Space List, pages are recycled during future operations.

A system transaction is complete when all the single requests and/or transactions in that system transaction are committed, and all the shadow pages involved in that system transaction have been written to disk. When a system transaction completes, Btrieve places the old original pages involved in that system transaction on the Free Space List.

If the system fails before a system transaction completes, Btrieve recovers all files involved in the failed system transaction.

Then, when it is first loaded (after you power up your system), Btrieve uses the yet unchanged original pages to return the files to the state they were in immediately prior to the failed system transaction. This recovery mechanism works regardless of whether the files have been flagged transactional.

Note: Client-based versions of Btrieve ignore a file's NetWare transaction tracking system (TTS) flag.

All changes made by a failed system transaction are lost. However, the file is left in a consistent state, allowing the operations to be attempted again after resolving the cause of the system failure.

Pre-imaging

For any Btrieve file using the pre-6.*x* format, Btrieve cannot apply shadow paging to guard against file corruption. Instead, it uses *pre-imaging*.

The effects of pre-imaging are roughly equivalent to those of shadow paging. However, pre-imaging is more time-consuming than shadow paging (another reason to consider converting your pre-6.*x* files to the 6.*x* format).

Unlike shadow paging, pre-imaging causes Btrieve to create a temporary pre-image *file* in the same directory where the original Btrieve data file resides. Btrieve creates this pre-image file the first time you modify a data file or access a data file during a transaction (except when the file is opened in read-only mode).

The temporary file has the same name as the file it is protecting, except that the filename extension is .PRE. For example, if you opened the ACCTID.DTA file, the temporary filename would be ACCTID.PRE. Btrieve uses this temporary file to track changes to the actual file during a request.

Do not create filenames with the extension .PRE because they will corrupt the recovery process.

Since the name of the pre-image file is the same as the Btrieve file (except for the extension), do not create multiple Btrieve files with the same names but different extensions in the same directory.

For example, do not use a naming scheme like INVOICE.HDR and INVOICE.DET for your Btrieve files. Otherwise, Btrieve tries to use INVOICE.PRE for both files and automatic recovery becomes impossible.

During the execution of a request or transaction that changes the file, Btrieve writes the image of the original pages to be changed to the pre-image file. Btrieve makes the requested modifications to the original pages where they reside in cache, then overwrites the original pages on disk with the new images from cache.

However, if partial changes made by a single request need to be rolled back, or if an Abort Transaction (21) operation is issued, Btrieve restores the original images from the pre-image file.

Important

Once Btrieve creates a pre-image file, do not delete it (except as preparation for converting a Btrieve file's format from 5.*x* to 6.*x*). Without the pre-image file, Btrieve cannot recover data after an abnormal termination.

For the purposes of data recovery, Btrieve uses internal system transactions for pre-6.*x* files just as it does for 6.*x* files. (System transactions are described in "Shadow Paging" on page 245.) Regardless of the file format, system transactions are completely transparent to an application unless a system failure occurs.

If the system fails before a system transaction completes, Btrieve recovers all the pre-6.*x* files involved in the failed system transaction.

When it is first loaded (after you power up your system), Btrieve uses the pre-image file to return the data files to the state they were in immediately prior to the failed system transaction. All changes made by a failed system transaction are lost. However, the file is left in a consistent state, allowing the operations to be attempted again after resolving the cause of the system failure.

Note If files from the network were open in a transaction at the time of the system failure, be sure that you are attached to the servers containing those files prior to starting Btrieve.

This recovery mechanism is usually successful. However, a very slim chance does exist that the recovery will be incorrect for Btrieve files that reside on a NetWare server.

This situation can occur if the NetWare disk-write optimizing algorithm causes the new image to be written to disk before the original page's image is written to the pre-image file. If this happens, and the system fails before the original page's image can actually be written to the pre-image file, the original data will not be recoverable except from backup.

You can override the NetWare disk-write optimizing algorithm by flagging your Btrieve files transactional.

Transaction Control

By defining explicit transactions within a program, you can force Btrieve to treat multiple Btrieve operations as an atomic unit—that is, Btrieve does not complete any of the operations in an explicit transaction unless it can successfully complete all of them.

Note For operations made inside a transaction involving a Btrieve file using the pre6.0 file format, changes are immediately visible to other users reading those files. With 6.*x*-format files, changes are not visible to other users until the End Transaction (20) operation completes.

If the system fails during an explicit transaction (which is automatically part of a system transaction), Btrieve removes (*rolls back*) all the operations performed during the system transaction.

To perform this automatic recovery procedure, Btrieve creates the file BTRIEVE.TRN whenever one or more files are opened in an explicit transaction. (BTRIEVE.TRN is located in the SYS:SYSTEM directory on the server unless otherwise specified during the Btrieve setup).

When Btrieve is loaded the first time after a system failure, it automatically checks BTRIEVE.TRN to see what files were involved in a transaction at the time the failure occurred. If BTRIEVE.TRN contains any filenames, Btrieve checks each of those files for consistency and returns the inconsistent files to their pre-transactional state, if necessary.

Note

Btrieve recovers each file listed in BTRIEVE.TRN with the exception of files that cannot be opened. To ensure that Btrieve can open all the files you were working on at the time of the system failure, you should log in to every server containing those files prior to restarting the application.

Occasionally, you may be unable to log in immediately to a server containing files with which you were working. In such a case, NetWare Btrieve cannot recover those files using transaction recovery (BTRIEVE.TRN). Btrieve will recover these files when someone logs into the server and opens the files. Therefore, individual files will not remain corrupted.

However, integrity between files cannot be guaranteed when files are recovered individually. Only transaction recovery can restore the integrity of the database. For this reason, if at all possible before you restart your application, log in to every server containing files you were working on at the time of the system failure.

As part of the recovery process, NetWare Btrieve prints a message to the window (or console) indicating which files are being rolled back. If a listed file cannot be opened, Btrieve prints a warning identifying the file and stating that it cannot be rolled back.

After recovery is completed, the first system transaction overwrites BTRIEVE.TRN.

Manual Data Recovery

You can use the Btrieve Maintenance utility command RECOVER to manually recover your data to a sequential file. The recovered data should reflect the status of the file before the interrupted operation began. (See “RECOVER” on page 159 in Chapter 5, “Using NetWare Btrieve Utilities” for information about using this command.)

If the system fails during a transaction, Btrieve rolls back all operations that completed after the Begin Transaction operation but before the system failure. As with all transactions, either all changes within t he transaction are made, or none are made.

After you recover the data, use the BUTIL commands CREATE or CLONE to make a new, empty, identical Btrieve file. (See “CREATE” on page 149 and “CLONE” on page 144 in Chapter 5 for detailed information about using these commands.)

Then use the sequential file with the BUTIL command LOAD to insert the records into the newly created Btrieve file. (See “LOAD” on page 156 in Chapter 5 for detailed information about using this command.)

Designing a Database

This section provides formulas and guidelines for designing a Btrieve database. The topics covered should help you set up security and reduce the amount of disk space your files use. This section discusses the following topics:

- "Determining Record and Page Size" on page 251
- "Estimating File Size" on page 257
- "Conserving Disk Space" on page 261
- "Setting Up Security" on page 266

Determining Record and Page Size

When you create a file, you must specify the logical record length and the file's page size. The following sections help you determine these values.

Logical Record Length

The *logical record length* is the number of bytes of fixed-length data you will allow a record to hold in the Btrieve file you are creating. To obtain this value, calculate how many bytes of data you need to store in the fixed-length portion of each record.

For example, Table 15 shows how the bytes in each fixed-length portion of the record are added together to obtain the logical record length.

Table 15
Calculating Logical Record Length for the Sample Record

Value	Number of Bytes for Value
Last name	25
First name	25
Middle initial	1
Employee ID	4
Street address	30

Table 15 *continued*

Calculating Logical Record Length for the Sample Record

Value	Number of Bytes for Value
City	25
State	11
Phone	12
Rate of pay	4
Department	24
Permanent/Temporary	1
Miscellaneous	0
Logical Record Length	162

The Miscellaneous portion of the record is variable length and contains notes about the employee (commendations, bonuses, and so on). Because Miscellaneous is variable length, it has no fixed length. Therefore, you do not need to reserve any bytes for its contents in each record.

If any record actually has information to store in Miscellaneous, that information is stored on variable pages and need not be included in the calculations for logical record length.

Note

For files that allow variable-length records, logical record length refers only to the fixed-length portion of the record.

Page Size

All pages in a Btrieve data file are the same size. Therefore, when you determine what size the pages should be in your file, you must choose a size that meets the following criteria:

- What page size is the smallest that will allow index pages to hold your largest key definition?
- What page size is the optimum size for data pages (which hold the fixed-length portion of records)?

Once you have these values, you can select a page size that best fits your file.

Minimum Page Size for Index Pages

To find the smallest page size that will still hold the largest key definition in your file, you must perform the following tasks:

Procedure

1. **Determine how many bytes the largest key in the file will require on an index page.**

   This value is the sum of the user-specified key length and the overhead that Btrieve requires to store the key. The user-specified key length is a given quantity. To calculate the number of overhead bytes required, perform the following substeps:

   - If the key does not allow duplicates, or if it allows *repeating* duplicates, add an additional 8 bytes to the user-specified key length and proceed to Step 2.
   - If the key allows *linked* duplicates, add an additional 12 bytes to the user-specified key length and proceed to Step 2.

2. **Multiply the value you obtained in Step 1 by 8. (Btrieve requires room for a minimum of 8 keys on a page.)**

3. **Add 12 to the value obtained in Step 2.**

4. **Select the first page size that is equal to or greater than the value you obtained in Step 3.**

Note

Remember that the page size you select in Step 4 must accommodate the size of any keys created after file creation.

Using the example record layout from Table 15, consider that your file will have 4 keys, as illustrated in Table 16.

Table 16

Keys Defined for the Sample Record

Key Name	Size in Bytes	Number of Segments	Duplicatable?
Last Name	25	1	Yes
Employee ID	4	1	No
Department	24	1	Yes
Permanent/Temporary	1	1	Yes

Given these values, you can calculate the minimum page size for your Btrieve file's index pages as follows:

- ◆ In performing Step 1 and its substeps, you can see that the largest key in the file (the last name key) will require a minimum of 25 bytes (its user-specified key length).

  In designing the file, you have decided that the last name key will allow duplicate values. At this point, you decide that those duplicate values should be stored as linked duplicates. Therefore, you would add 12 bytes for overhead to the user-specified key length of 25, giving you a total of 37 bytes.

- ◆ Performing Step 2, you multiply 37 by 8, giving you a total of 296 bytes.

- ◆ Performing Step 3, you add 12 bytes to the 296 bytes from Step 2, giving you a total of 308 bytes.

- ◆ Performing Step 4, you select the first page size that is equal to or greater than 308 bytes.

  Because the minimum Btrieve page size is 512 bytes, you would select 512.

Although you have not eliminated any possible page sizes in this example, you must verify the minimum page size for index pages before choosing the optimum page size for data pages. Failure to do so may result in selecting a data page size that is incompatible with the allowable index page size.

Optimum Page Size for Data Pages

Before you can determine the optimum size of your Btrieve file's data pages, you must first calculate the file's *physical record length*. The physical record length is the sum of the logical record length and the *overhead* required to store a record on a data page of the Btrieve file.

Btrieve always stores a minimum of 2 bytes of overhead information in every record (as a usage count for that record). Btrieve also stores an additional number of bytes in each record, depending on how you define the records in your Btrieve file.

Table 17 shows how many bytes of overhead you must add to the logical record length to obtain the physical record length (based on how you define the records for your Btrieve file).

Table 17
Amount of Overhead Added to the Logical Record Length of the Sample Record

Type of Overhead	Bytes
Record usage count	2 (only for files using Btrieve v6.*x* format
Linked-duplicatable keys	8 (per key)
Reserved duplicate pointers	8 (per pointer)
Variable-length record pointers	4 (if file allows variable-length records)
Blank truncation	2 (if file allows blank truncation but does not use VATs), or 4 (if file allows blank truncation and uses VATs)
Record length	4 (if file uses VATs)

Important

When calculating the number of bytes for duplicatable keys, include only 8 bytes for each linked-duplicatable key. Btrieve does not allocate duplicate pointer space for keys defined as repeating duplicatable at creation time.

Using the sample record defined in Table 8 on page 198, assume that you are creating a Btrieve file with the following attributes, or characteristics:

- A logical record length of 162 bytes.

- The variable-length portion of the record is blank-truncated. (Blank truncation is discussed later in this appendix.)

- The file has three linked-duplicatable keys and does not use VATs.

To obtain the physical record length for this Btrieve file, you would add the values shown in Table 18.

Table 18

Obtaining the Physical Record Length of the Sample File

Logical record length	162
Record usage count	2
3 linked-duplicatable keys (3*8)	24
Variable-length record pointer	4
Blank truncation	0
Physical record length	194

Remember that for files with variable-length records, the logical record length refers only to the fixed-length portion of the record.

Note For files with compressed records, the data page format is different. (Compressed records are discussed in “Data Compression” on page 263.) A compressed record’s entry on a data page consists of 7 bytes of overhead information, pointers for duplicate keys, reserved duplicate pointers (if any), and a 4-byte record-length field (if the file uses VATs). The record’s user-defined data is compressed and stored on variable pages.

Using the physical record length that you calculated in the preceding paragraphs, you now can determine the file’s optimum page size for data pages.

Btrieve stores as many records as possible in each data page of the Btrieve file. However, it will not break the fixed-length portion of a record across pages. Also, in each data page, Btrieve stores six bytes of overhead information. You must account for this additional overhead when determining the data page size.

A Btrieve file will contain unused space if the page size you choose, minus six bytes (for overhead information), is not an exact multiple of the physical record length.

To optimize your file's use of disk space, select a page size that can buffer your records with the least amount of unused space. Page size must always be a multiple of 512 bytes, up to 4,096 bytes. Larger page sizes usually result in more efficient use of disk space.

Consider the example from Table 18, in which the physical record length added up to 194 bytes. Table 19 shows how many records can be stored on a page and how many bytes of unused space will remain on a page for each possible page size.

Table 19
Optimizing the Sample File's Use of Disk Space

Page Size	Records per Page	Unused Bytes	
512	2	118	(512 – 6 – (2 * 194))
1024	5	48	(1024 – 6 – (5 * 194))
1536	7	172	(1536 – 6 – (7 * 194))
2048	10	102	(2048 – 6 – (10 * 194))
2560	13	32	(2560 – 6 – (13 * 194))
3072	15	156	(3072 – 6 – (15 * 194))
3584	18	86	(3584 – 6 – (18 * 194))
4096	21	16	(4096 – 6 – (21 * 194))

As the table indicates, if you select a page size of 512, only 2 records can be stored per page and 118 bytes of each page will be unused. However, if you select a page size of 4,096, 21 records can be stored per page and only 16 bytes of each page will be unused.

Note Btrieve requires that each index page in the file be large enough to hold at least eight keys. If an index page is not large enough, you must either increase the file's page size or decrease the key length. For more information, see "Minimum Page Size for Index Pages" on page 253.

Estimating File Size

You can estimate the number of pages, and therefore the number of bytes required to store a Btrieve file, with the formulas described in the following paragraphs. However, when using the formulas, keep in mind that at any given moment, they only approximate file size because of the way Btrieve dynamically manipulates pages.

Note

The following discussion and the formulas for determining file size do not apply to files that use data compression, because the record length for those files depends on the number of repeating characters in each record.

While the formulas are based on the maximum storage required, they assume that only one Btrieve task is updating or inserting records into the file at a time. File size increases if more than one task updates or inserts records into the file during simultaneous concurrent transactions.

The formulas also assume that no records have yet been deleted from the file. Even if you delete half the records in a file, the file remains the same size. Btrieve does not deallocate the pages that were occupied by the deleted records. Rather, Btrieve re-uses them as new records are inserted into the file (before allocating new pages).

If the final outcome of your calculations contains a fractional value, round the number to the next highest whole number.

To estimate the size of your Btrieve file, perform the following steps:

Procedure

1. **Calculate the number of data pages, using the following formula:**

   Number of data pages =

   Number of records /
   ((Page size - 6) / (Physical record length))

2. **To find the physical record length, use the following formula:**

   Physical record length =

   Number of bytes of data in the fixed-length portion of the record +

   2 bytes for the usage count +

   (8 * Number of keys at CREATE time allowing duplicates) +

   (8 * Number of reserved duplicate pointers) +

   4 bytes if the record is a variable-length record +

2 bytes if blank truncation is allowed (but the file does *not* use VATs), or

4 bytes if blank truncation is allowed and the file *does* use VATs) +

4 bytes if the file uses VATs (regardless of whether blank truncation is allowed)

Important

When calculating the number of bytes for duplicatable keys, include only 8 bytes for each linked-duplicatable key. Btrieve does not allocate duplicate pointer space for keys defined at creation time as repeating duplicatable.

3. **Calculate the number of index pages for each defined key using one of the following formulas:**

   - For each key that does *not* allow duplicates or that allows repeating-duplicatable keys:

   Number of index pages =

   (Number of records /
   ((Page size - 12) / (Key length + 8))

) * 2

   - For each key that allows linked-duplicatable keys:

   Number of index pages =

   (Number of unique key values /
   ((Page size - 12) / (Key length + 12))

) * 2

   The B-tree index structure guarantees at least 50 percent usage of the index pages. Therefore, the index page calculations multiply the minimum number of index pages required by two to account for the maximum size.

4. **If your file contains variable-length records, calculate the number of variable pages in the file and add that number to the sum from the preceding steps.**

   To do so, use the following formula:

   Number of variable pages =

   (Total number of records in the file) /

   (Average number of records whose variable-length portion fits on a single page)

Note You can gain only a very rough estimate of the number of variable pages due to the difficulty in estimating the average number of records whose variable-length portion will fit on the same page.

5. **To the sum obtained in the preceding steps, add the following:**

   1 page for each alternate collating sequence page used (if any)

   1 page for a referential integrity (RI) page if the file has RI constraints

   This new sum represents the estimated total number of logical pages that the file will contain.

6. **Calculate the number of PAT pages, and add that number to the estimated number of logical pages from the preceding step.**

   Every file has a minimum of two PAT pages. However, to calculate the number of PAT pages in a file, use the following formula:

   Number of PAT pages =

   ((Sum of pages in Steps 1 through 3) * 4) /
   ((Page size - 8 bytes for overhead) * 2)

7. **To the sum obtained in the preceding step, add 2 pages for the FCR pages.**

8. **Finally, add the estimated size of the pool of unused pages in the file. Btrieve uses the pool for shadow paging.**

   To calculate the size of the pool, use the following formula:

   Size of the pool of unused pages =

   ```
   (Number of keys + 1)
   ```

   This formula applies if Btrieve tasks will execute insert, update, and delete operations only outside transactions. If tasks will be executing these operations inside transactions, multiply the average number of insert, update, and delete operations expected in the transactions times the non-transactional figure determined by the formula.

   Similarly, you must further increase the estimated size of the pool of unused pages if tasks will be executing simultaneous concurrent transactions.

9. **Having calculated the number of pages needed by the file, use the following formula to calculate the maximum number of bytes required to store the file:**

   File size in bytes = Total file pages * Page size

Note

Add an additional 4,096 bytes to the size of the file if the file uses more than 30 PAT pages. Btrieve uses these additional bytes internally.

Conserving Disk Space

Btrieve provides several features that allow you to conserve disk space and improve system performance. These features include file preallocation, blank truncation, data compression, and index compaction.

File Preallocation

Preallocation guarantees that disk space will be available when Btrieve needs it. Btrieve allows you to preallocate up to 65,535 *pages* to a file when you create a data file.

Table 20 shows the maximum number of *bytes* Btrieve allocates for a Btrieve file of each possible page size (assuming you allocate a full 65,535 pages of disk space).

Table 20
Maximum Number of Bytes Allocated per File by Page Size

Page Size	Disk Space Allocated	
512	33,553,920	(512 * 65,535)
1024	67,107,840	(1024 * 65,535)
1536	100,661,760	(1536 * 65,535)
2048	134,215,680	(2048 * 65,535)
2560	167,769,600	(2560 * 65,535)
3072	201,323,520	(3072 * 65,535)
3584	243,877,440	(3584 * 65,535)
4096	268,431,360	(4096 * 65,535)

If not enough space exists on the disk to preallocate the number of pages you specify, Btrieve returns Status Code 18 (Disk Full) and does not create the file.

The speed of file operations can be enhanced if a data file occupies a *contiguous* area on the disk. The increase in speed is most noticeable on very large files.

To preallocate contiguous disk space for a file, the device on which you are creating the file must have the required number of bytes of contiguous free space available. Btrieve preallocates the number of pages you specify, whether or not the space on the disk is contiguous.

Use the formulas described earlier in this appendix to determine the number of data and index pages the file requires. You should round any remainder from this part of the calculation to the next highest whole number.

When you preallocate pages for a file, that file actually occupies that area of the disk. No other data file can use the preallocated area of disk until you delete or replace that file.

As you insert records, Btrieve uses the preallocated space for data and indexes. When all the preallocated space for the file is in use, Btrieve expands the file as new records are inserted.

When you issue a Btrieve Stat (15) operation, Btrieve returns the difference between the number of pages you allocated at the time you created the file and the number of pages that Btrieve has currently in use. This number is always less than the number of pages you specify for preallocation because Btrieve considers a certain number of pages to be in use when a file is created, even if you have not inserted any records.

Once a file page is in use, it remains in use even if you delete all the records stored on that page. Thus, the number of unused pages that the Stat operation returns never increases. When you delete a record, Btrieve maintains a list of free space in the file and reuses the available space when you insert new records.

Even if the number of unused pages returned by the Stat operation is zero, the file might still have free space available. The number of unused pages can be zero if one of the following is true:

- You did not preallocate any pages to the file.

- All the pages that you preallocated were in use at one time or another.

Blank Truncation

Blank truncation can be used only with variable-length records. When you define a file that allows variable-length records, you can specify that Btrieve use a blank truncation method for storing the records in order to conserve disk space.

If you choose to truncate blanks, Btrieve does not store any trailing blanks in the variable-length portion of the record when it writes the record to the file. Blank truncation has no effect on the fixed-length portion of a record. Btrieve does not remove blanks that are embedded in the data.

When you read a record that contains truncated trailing blanks, Btrieve expands the record to its original length. The value Btrieve returns in the data buffer length parameter includes any expanded blanks.

Blank truncation adds either 2 bytes or 4 bytes of overhead to the physical size of the record (stored with the fixed-record portion): 2 if the file does not use VATs, 4 if it does.

Data Compression

When you create a Btrieve file, you can specify if you want Btrieve to compress the data records when it stores them in the file. Data compression can result in a significant reduction of the space needed to store records that contain many repeating characters. Btrieve compresses 5 or more of the same contiguous characters into 5 bytes.

You should consider using Btrieve data compression in the following circumstances:

- The records to be compressed are structured so that the benefits of using data compression are maximized.
- The need for better disk utilization outweighs the possible increased processing and disk access times required for compressed files.
- The computer running Btrieve can supply the extra memory used by Btrieve for compression buffers.

When you perform record I/O on compressed files, Btrieve uses a compression buffer to provide a block of memory for the record compression and expansion process. To ensure sufficient memory to compress or expand a record, Btrieve requires enough buffer space to store twice the length of the longest record your Btrieve task inserts into the compressed file.

This requirement can have an impact on the amount of free memory left in the computer after Btrieve loads. For example, if the longest record your task writes or retrieves is 2,000 bytes long, Btrieve requires 4,000 extra bytes of memory to compress and expand that record.

Note If your file uses VATs (see “Storing Variable-Length Records: Variable Tails and VATs” on page 209), Btrieve’s requirement for buffer space is the product of 16 times the file’s page size.

Because the final length of a compressed record cannot be determined until the record is written to the file, Btrieve always creates a compressed file as a variable-length record file. In the data page, Btrieve stores either 7 bytes (if the file does not use VATs) or 11 bytes (if it does), plus an additional 8 bytes for each duplicate key pointer. Btrieve then stores the record on the variable page.

Since the compressed images of the records are stored as variable-length records, individual records may become fragmented across several file pages if your Btrieve task performs frequent insertions, updates, and deletions. The fragmentation can result in slower access times because Btrieve may need to read multiple file pages to retrieve a single record.

The data compression option is most effective when each record has the potential for containing a large number of repeating characters. For example, a record may contain several fields, all of which may be initialized to blanks by your Btrieve task when it inserts the record into the file. Compression is more efficient if these fields are grouped together in the record, rather than being separated by fields containing other values.

To use data compression, the file must have been created with the compression flag set. Note that key-only files do not allow compression.

Index Balancing

Btrieve allows you to further conserve disk space by employing *index balancing*. By default, Btrieve does not use index balancing, so that each time a current index page is filled, Btrieve must create a new index page. When index balancing is active, Btrieve can frequently avoid creating a new index page each time a current index page is filled.

Index balancing forces Btrieve to look for available space on adjoining index pages. If space is available on one of those pages, Btrieve moves keys from the full index page to the page with free space.

The balancing process not only results in fewer index pages but also produces more densely populated indexes, better overall disk usage, and faster response on most read operations.

If you add keys to the file in sorted order, index page usage increases from 50 percent to nearly 100 percent when you use index balancing. If you add keys randomly, the minimum index page usage increases from 50 percent to 66 percent.

On write operations, the balancing logic requires Btrieve to examine more pages in the file and might possibly require more disk I/O. The extra disk I/O slows down file updates. Although the exact effects of balancing indexes vary in different situations, using index balancing typically degrades performance on write operations by about 5 to 10 percent.

Btrieve allows you to fine-tune your Btrieve environment by offering two levels of index balancing.

- If you specify the index balancing configuration option during setup, when Btrieve is started, it applies index balancing to every key in every Btrieve file.

  For a description of how to specify the index balancing configuration option, see “Perform Index Balancing” on page 69 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

- You can also designate that only the keys in specific files are to be index balanced.

  To do so, set bit 5 (0x10) of the affected file’s file flags at creation time. If the index balancing configuration option is off when Btrieve is started, Btrieve applies index balancing only to files with bit 5 of the file flags set.

  Btrieve ignores bit 5 in all files’ file flags if the index balancing configuration option was on when Btrieve was started. In this situation, Btrieve applies index balancing to every Btrieve file.

Btrieve files remain compatible regardless of whether index balancing is active. Also, you do not have to specify index balancing to access files that contain balanced index pages.

Setting Up Security

Btrieve provides two methods of setting up file security:

- Assigning an owner name to the file
- Opening the file in exclusive mode

Owner Names

Btrieve allows you to restrict access to a file by specifying an owner name, using either the Maintenance utility command SETOWNER or through issuing the Set Owner (29) operation in an application.

Once you assign an owner name to a file, Btrieve requires that name to be specified for all future modifications to the file. This prevents any unauthorized access or changing of a file’s contents by users or applications that do not have access to the owner name.

You can restrict access to the file in several ways:

- Users can have read-only access *without* an owner name.
- Users can be required to supply an owner name for any access mode.

In the first situation, a Btrieve task may read the file without being required to specify the owner name. However, neither a user nor a task can change the file’s contents without specifying the owner name.

In the second situation, Btrieve restricts all access to the file unless the owner name is specified.

As a complement to these restrictions, Btrieve allows you to remove ownership restrictions from a file if you know the owner name assigned to it. To do so, use the Btrieve Maintenance command CLOWNER, or the application can issue a Clear Owner (30) operation.

When you assign an owner name, you may also request that Btrieve encrypt the data in the disk file using the owner name as the encryption key. Encrypting the data on the disk ensures that unauthorized users cannot examine your data by using a debugger or a file dump utility.

Since encryption requires additional processing time, select this option only if data security is important in your environment.

Exclusive Mode

If you want to limit access to a file to a single task, you can specify that Btrieve open the file in exclusive mode. When a task opens a file in exclusive mode, no other task can open the file until the task that opened the file in exclusive mode closes it.

appendix B Description Files

A *description file* is an ASCII file containing information that the Btrieve Maintenance utility commands CREATE, INDEX, and SINDEX need to perform their operations.

Description files contain one or more *elements*. An element consists of a keyword, followed by an equal sign (=), followed by a value (with no space). Each element in the description file corresponds to a particular characteristic of a Btrieve® file or key definition.

The sections in this appendix discuss the following topics:

- “Rules for Description Files” on page 269
- “Description File Example” on page 271
- “Description File Elements” on page 274

Rules for Description Files

Use the following rules when creating a description file.

- Enter elements in either uppercase or lowercase letters, as in the following example:

  ```
  TYPE=FLO
  ```

  or

  ```
  type=flo
  ```

- Separate elements from each other with a separator (blank space, tab, or carriage return/line feed):

  ```
  record=4000 key=24
  ```

- Specify the description file elements in the proper order. Table 21 on page 274 presents the elements in the appropriate order.

Note The order of the elements required for the CREATE, INDEX, and SINDEX commands is the same. However, these commands do not all require the same elements.

- Address all element dependencies. For example, consider the following All Segments (null key) element:

```
nullseg=allsegs
```

  If you specify this element in the description file, you must also include the Null Key Value element.

- Define as many keys as you specify with the Key Count element. The following example specifies 12 keys for the file:

```
key=12
```

  In this case, you must define 12 keys in the description file, each consisting of one or more segments.

- For a key with multiple segments, you must define the following elements for each key segment:
  - Key Position
  - Key Length
  - Duplicate Key Values
  - Modifiable Key Values
  - Key Type
  - Alternate Collating Sequence

  The Descending Sort Order element is optional for each segment.

- If any key in the file uses an alternate collating sequence, include either an alternate collating sequence filename or a country ID and code page ID. You can include this information as either the last element of the key segment or the last element in the description file.

- If a description file element is optional, you can omit it from the description file.

- Make sure the description file contains no text formatting characters. Some word processors embed formatting characters in a text file.

Description File Example

The sample description file shown in Figures 29, 30, and 31 creates a Btrieve file with a page size of 512 bytes and 2 keys. The fixed-length portion of the record is 98 bytes long. The file allows variable-length records but does not use blank truncation.

The file uses data compression, allows for Variable-tail Allocation Tables (VATs), and has the free space threshold set to 20 percent. Btrieve preallocates 100 pages, or 51,200 bytes, when it creates the file. The file has two keys: Key 0 and Key 1. Key 0 is a segmented key with two segments.

Figure 28
Sample Description File Using Alternate Collating Sequence Filename

```
record=98 variable=y truncate=n compress=y       File
key=2 page=512 allocation=100 replace=n          Specifications
fthreshold=20 vats=y

position=1 length=5 duplicates=y                 Key 0
modifiable=n type=string alternate=y             Segment 1
nullkey=allsegs value=20 segment=y

position=6 length=10 duplicates=y                Key 0
modifiable=n type=string alternate=y             Segment 2
nullkey=allsegs value=20 segment=n

position=16 length=2 duplicates=n
modifiable=y type=numeric descending=y           Key 1
alternate=n nullkey=n segment=n

name=C:/upper.alt
```

In Figure 28, an alternate collating sequence filename (UPPER.ALT) is specified for Key 0.

If you specify y for the Alternate Collating Sequence element for a key, you must supply an alternate collating sequence filename or a country ID and code page ID.

In Figure 29, a country ID and code page ID are specified (`countryid=-1` and `codepageid=-1`). If you specify the `name=`*`sequenceFile`* element (or the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) at the end of the description file, Btrieve uses it as the default alternate collating sequence.

That is, if you specify `alternate=y` for a given key but do not include a `name=`*`sequenceFile`* element (or the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) for that key, Btrieve uses the `name=`*`sequenceFile`* element (or the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) specified at the end of the description file.

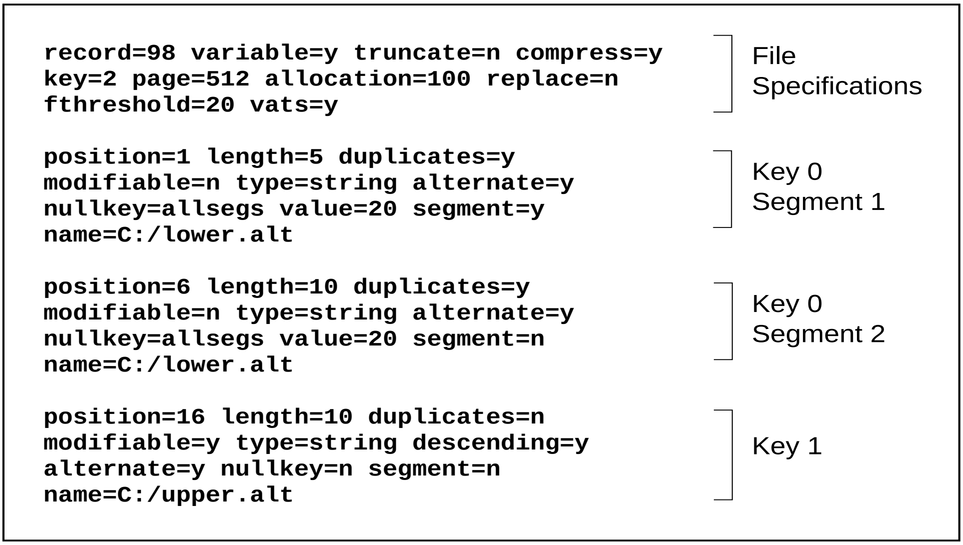
Figure 29

Sample Description File Using Alternate Collating Sequence ID

```
record=98 variable=y truncate=n compress=y      File
key=2 page=512 allocation=100 replace=n         Specifications
fthreshold=20 vats=y

position=1 length=5 duplicates=y                Key 0
modifiable=n type=string alternate=y            Segment 1
nullkey=allsegs value=20 segment=y

position=6 length=10 duplicates=y               Key 0
modifiable=n type=string alternate=y            Segment 2
nullkey=allsegs value=20 segment=n

position=16 length=2 duplicates=n
modifiable=y type=numeric descending=y          Key 1
alternate=n nullkey=n segment=n

countryid=-1  codepageid=-1
```

In Figure 30, a different alternate collating sequence filename is specified for Key 0 and Key 1. If you want to use different alternate collating sequences for different keys, you must specify the `name=`*`sequenceFile`* element (or the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) for each key that uses a different alternate collating sequence.

Figure 30
Sample Description File Using Alternate Collating Sequence Filename on a File Segment



```
record=98 variable=y truncate=n compress=y
key=2 page=512 allocation=100 replace=n
fthreshold=20 vats=y

position=1 length=5 duplicates=y
modifiable=n type=string alternate=y
nullkey=allsegs value=20 segment=y
name=C:/lower.alt

position=6 length=10 duplicates=y
modifiable=n type=string alternate=y
nullkey=allsegs value=20 segment=n
name=C:/lower.alt

position=16 length=10 duplicates=n
modifiable=y type=string descending=y
alternate=y nullkey=n segment=n
name=C:/upper.alt
```

Different segments of the same key cannot have different alternate collating sequences.

You can specify only one alternate collating sequence per key, and you must provide an alternate collating sequence filename (or values for the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) for each segment of the key.

The filename for the `name=`*`sequenceFile`* element (or values for the `countryid=`*`nnn`* and `codepageid=`*`nnn`* elements) must be the same for each segment. In Figure 30, the filename specified for each segment in Key 0 is LOWER.ALT. The filename specified for Key 1 is UPPER.ALT.

For more information, see “Alternate Collating Sequence” on page 216 in Appendix A, “Btrieve Concepts.”

Description File Elements

Table 21 lists the description file elements in the order in which they must appear in the description file.

For each element, Table 21 specifies the required format, the range of acceptable values, and the associated commands. An asterisk (*) after the element name indicates that the element is optional. Descriptions of the individual elements follow the table.

Table 21
Summary of Description File Elements

Element	Format	Range	Command
Comment Block	/*. . .*/	5,120 bytes	CREATE
Record Length	record=*nnnn*	4–4,088	CREATE
Variable-Length Records	variable=<y|n>	N/A	CREATE
Reserved Duplicate Pointer*	dupkey=<nnn>	1–119	CREATE
Blank Truncation*	truncate=<y|n>	N/A	CREATE
Data Compression*	compress=<y|n>	N/A	CREATE
Key Count	key=*nnn*	0–118	CREATE
Page Size	page=*nnnn*	512–4,096	CREATE
Page Preallocation*	allocation=*nnnnn*	1–65,535	CREATE
Replace Existing File*	replace=<y|n>	N/A	CREATE
Include Data*	data=<y|n>	N/A	CREATE
Free Space Threshold*	fthreshold=<10|20|30>	N/A	CREATE
Variable-tail Allocation Tables (VATs)*	vats=<y|n>	N/A	CREATE
Balanced Index*	balance=<y|n>	N/A	CREATE
Key Position	position=*nnnn*	1–4,088	CREATE, INDEX, SINDEX

Table 21 *continued*

Summary of Description File Elements

Element	Format	Range	Command
Key Length	length=*nnn*	key type limit	CREATE, INDEX, SINDEX
Duplicate Key Values	duplicates=<y|n>	N/A	CREATE, INDEX, SINDEX
Modifiable Key Values	modifiable=<y|n>	N/A	CREATE, INDEX, SINDEX
Key Type	type=*validBtrieveKeyType*	N/A	CREATE, INDEX, SINDEX
Descending Sort Order*	descending=<y|n>	N/A	CREATE, INDEX, SINDEX
Alternate Collating Sequence*	alternate=<y|n>	N/A	CREATE, INDEX, SINDEX
Case-Insensitive Key*	caseinsensitive=<y|n>	N/A	CREATE, INDEX, SINDEX
Repeating Duplicates*	repeatdup=<y|n>	N/A	CREATE, SINDEX
Any Segment Null*	nullkey=anyseg	N/A	CREATE, INDEX, SINDEX
All Segments Null*	nullkey=allsegs | n	N/A	CREATE, INDEX, SINDEX
Null Key Value	value=*nn*	1-byte hex	CREATE, INDEX, SINDEX
Segmented Key	segment=<y|n>	N/A	CREATE, INDEX, SINDEX
Alternate Collating Sequence Filename/ID	name=*sequenceFile or* countryid=*nnn* and codepageid=*nnn*	valid path *or* values valid to operating system or –1 (default)	CREATE, INDEX, SINDEX

Comment Block

Format: `/* . . .*/`

Range: 5,120 bytes

The optional Comment Block element provides a section at the beginning of the description file in which you can enter comments about that file, or any other type of relevant information.

You can enter up to 5,120 bytes of data in this section.

Record Length

Format: `record=`*`nnnn`*

Range: 4 through 4,088 bytes

Command: CREATE

The Record Length element defines the logical data record length in bytes. A record represents a set of logically associated items in a Btrieve file. Generally, a record is the unit transferred between an application and Btrieve in a single operation.

There is no inherent restriction on the number of records allowed in a Btrieve file. A record can consist entirely of a fixed-length portion (a fixed-length record), or it can consist of a fixed-length portion followed by a variable-length portion (a variable-length record).

Note All the keys defined for the file must be located within the fixed-length portion of the record.

The logical record length is the number of bytes of fixed-length data you allow a record in the specified Btrieve file to hold. To determine this value, calculate how many bytes of data you need to store in the fixed-length portion of each record.

- For fixed-length records, the value for this element should correspond to the length of the data buffer parameter that performs operations on the file.

- For variable-length records, the value for this element should correspond to the fixed-length portion of the record.

In a file that uses fixed-length records, the length of each record in that file must be the same. The maximum length of a fixed-length record depends on the physical page size and the number of duplicate keys you define for the file.

Btrieve allows a fixed-length record to contain up to 4,088 bytes (4,096 minus 6 bytes for page overhead information and 2 bytes for the record usage count).

The data record length must be at least 4 bytes and must be large enough to contain all the keys defined for the file. The record length (including its duplicate key overhead and usage count overhead) plus 6 bytes must not exceed the file's page size.

For more information about records, see "Records" on page 207 in Appendix A, "Btrieve Concepts."

Variable-Length Records

Format: `variable=<y|n>`
Range: Not applicable
Command: CREATE

The Variable-Length Records element specifies whether the file will contain variable-length records. If a record is variable length, the maximum fixed-length record is decreased by 4 bytes.

Variable-length records store data of indeterminate length. When you create a Btrieve file, you can specify that you want the file to use variable-length records.

A variable-length record has a fixed-length portion and a variable-length portion. While the fixed-length portion of all variable-length records in a file must be the same size, the variable-length portion can vary. This means that the overall length of each variable-length record in the file can vary.

When you create a Btrieve file that uses variable-length records, you specify the length in bytes of the fixed-length portion of each record. This fixed-length amount is the minimum length of each record; you do not define the maximum record length.

Beginning with Btrieve v6.1, the maximum length of variable-length records is limited only by Btrieve's file size limit: 4 gigabytes. However, an application's ability to insert, update, and retrieve long records is influenced by several factors.

Specify y if you want the file to allow variable-length records. Otherwise, specify n.

For more information, see “Variable-Length Records” on page 207 in Appendix A.

Reserved Duplicate Pointer

Format: `dupkey=`*`nnn`*

Range: 1 through 119

Command: CREATE

The Reserved Duplicate Pointer element is optional. It specifies the number of duplicate key pointers to preallocate for the file when it is created. Each reserved duplicate pointer adds 8 bytes of extra storage space to the fixed record length.

You can use these reserved pointers when you add keys that allow duplicates after the file is created, and if you do not specify `y` for the Repeating Duplicates element.

Blank Truncation

Format: `truncate=<y|n>`

Range: Not applicable

Command: CREATE

The Blank Truncation element is optional. It specifies whether Btrieve performs blank truncation on variable-length records.

When you define a file that allows variable-length records, you can specify that Btrieve use a blank truncation method for sorting the records in order to conserve disk space.

The truncate element has an effect only if you also specify y for the Variable-Length Records element. Btrieve ignores blank truncation if you specify data compression.

Specify y if you want Btrieve to use blank truncation. Otherwise, either specify n or omit this element from your description file.

For more information, see “Blank Truncation” on page 263 in Appendix A.

Data Compression

Format: `compress=<y|n>`

Range: Not applicable

Command: CREATE

The Data Compression element is optional. It specifies whether Btrieve performs data compression on records that are inserted into the file.

If you specify data compression when you create a Btrieve file, Btrieve compresses the file's records before inserting or updating them, and uncompesses the records when it retrieves them.

Because the final length of a compressed record cannot be determined until the record is written to the file, Btrieve always creates a compressed file as a variable-length record file.

Note A compressed fixed-length-record file differs from a compressed variable-length-record file only in that Btrieve prevents insert and update operations from producing a record longer than a compressed fixed-length-record file's specified record length.

Since the compressed images of the records are stored as variable-length records, individual records may become fragmented across several file pages if an application performs frequent insertions, updates, and deletions. The fragmentation can result in slower access times because Btrieve may need to read multiple file pages to retrieve a single record.

However, data compression can also result in a significant reduction of the space needed to store records that contain many repeating characters. Btrieve compresses 5 or more of the same continguous characters into 5 bytes.

Btrieve does not use data compression with key-only files (nor is there blank truncation with key-only files).

Specify y if you want Btrieve to perform data compression. Otherwise, either specify n or omit this element from your description file.

For more information, see "Data Compression" on page 263 in Appendix A.

Key Count

Format: `key=`*`nnn`*

Range: 0 through 118 (see Table 22)

Command: CREATE

The Key Count element specifies the number of keys to be defined in the file. If you specify a value of 0 for this element, Btrieve creates a data-only file. If you specify a value greater than 0, Btrieve creates either a standard file or a key-only file, depending on the value you specify for the Include Data element.

The file's page size limits the amount of key segments a file can have. Table 22 shows the maximum key count values for each possible page size. This value represents the number of key segments, not the number of keys.

There may actually be more segment definitions than the key count.

For example, assume a file contains three keys: key 0 has 2 segments, key 1 has 4 segments, and key 2 has 2 segments. In this example, the file has a page size of 512, the value specified for the Key Count element is 3, and the maximum number of key segments is 8.

Table 22
Key Segment Count Values

Page Size (in bytes)	Maximum Number of Key Segments
512	8
1,024	23
1,536	24
2,048–3,584	54
4,096	119

For more information, see "Keys" on page 211 in Appendix A.

Page Size

Format: page=*nnnn*

Range: 512 through 4,096 bytes

Command: CREATE

The Page Size element specifies the physical page size (in bytes) for the file. You can specify any multiple of 512, up to 4,096.

Note

NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size. However, NetWare Btrieve does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

All pages in a Btrieve data file are the same size. Therefore, when you determine what size the pages should be in your file, you must choose a size that meets the following criteria:

- What page size is the smallest that will allow index pages to hold your largest key definition?

- What page size is the optimum size for data pages?

Once you have these values, you can select a page size that best fits your file.

Hint For optimum performance, set the page size to an even multiple of 512, 1,024, 2,048, or 4,096 bytes.

For information about page types, see “File Components” on page 200 in Appendix A. For information about setting the minimum page size for index pages and the optimum page size for data pages, see “Determining Record and Page Size” on page 251 in Appendix A.

Page Preallocation

Format: `allocation=`*`nnnnn`*

Range: 1 through 65,535

Command: CREATE

The Page Preallocation element is optional. It specifies the number of pages to preallocate to the file. (Btrieve allows you to preallocate up to 65,535 pages when you create a data file.) Preallocation guarantees that disk space will be available when Btrieve needs it.

You can enhance the speed of file operations if a data file occupies a contiguous area on the disk. The increase in speed is most noticeable on very large files.

To preallocate contiguous disk space for a file, the device on which you are creating the file must have the required number of bytes of contiguous free space available. Note, however, that Btrieve preallocates the number of pages you specify whether or not the space on the disk is contiguous.

Hint

Use the formulas described in “Determining Record and Page Size” on page 251 in Appendix A to determine the number of data and index pages the file requires.

If you do not want to preallocate any pages, either specify n or omit this element from your description file.

For more information, see “Page Size” on page 253 and “File Preallocation” on page 261 in Appendix A.

Replace Existing File

Format: `replace=<y|n>`

Range: Not applicable

Command: CREATE

The Replace Existing File element is optional.

If you do not want to create a new, empty file over an existing Btrieve file of the same name, specify n. If you want to replace an existing Btrieve file with a new, empty file of the same name, either specify y or omit this element from your description file.

Include Data

Format: `data=<y|n>`

Range: Not applicable

Command: CREATE

The Include Data element is optional. It specifies the type of file (key-only, standard, or data-only) that the utility creates.

Key-only files contain only File Control Record (FCR) pages followed by a number of Page Allocation Table (PAT) pages. If a key-only file has an alternate collating sequence, then it may also have an alternate collating sequence page.

Similarly, if anyone has used Scalable SQL to define referential integrity (RI) constraints on the file, then the file may contain one or more variable pages.

Note

A standard Btrieve v6.*x* file contains two FCR pages followed a number of PAT pages, index pages, data pages, and possibly variable pages. For additional information, see "File Types" on page 203 in Appendix A.

In a key-only file, the entire record is stored with the key, so no data pages are required. Key-only files are useful when your records contain a single key, and that key takes up most of each record. Another common use for a key-only file is as an external index for a standard Btrieve file.

To create a key-only file, specify n. To create a standard file, either specify y or omit the element from the description file. To create a data-only file, specify y and set the Key Count element to 0.

For more information, see "Key-Only Files" on page 204 in Appendix A.

Free Space Threshold

Format: `fthreshold=<5|10|20|30>`

Range: 5 through 30 percent

Command: CREATE

Btrieve stores the variable-length portions of records on their own pages (called *variable pages*), separate from the fixed-length portion of the record (which is stored on a data page). Btrieve maintains the Free Space List for variable pages.

The Free Space List indicates which variable pages contain the same or more free space than that specified by the Free Space Threshold file specification. (For additional information, see “Free Space List” on page 210 in Appendix A.)

You can specify a value for the Free Space Threshold element when you create the file. This value is expressed as a percentage and tells Btrieve how much free space must remain on a variable page in order for that page to appear on the Free Space List.

When Btrieve adds new variable-length records to the file, it uses pages in the Free Space List before using new variable pages.

You can specify any two-digit number for this element, and the utility rounds it to 10, 20, or 30.

Btrieve uses a default free space threshold of 5 percent if either of the following is true:

- You specify a value less than 10.
- You do not specify a value here but have specified y for the Variable-Length Records element.

Note If the Btrieve file does not allow variable-length records, do not include this element in the description file.

Variable-tail Allocation Tables (VATs)

Format: vats=<y|n>

Range: Not applicable

Command: CREATE

The Variable-tail Allocation Table (VAT) element is optional. Btrieve v6.1*x* allows an application to create Btrieve files that contain structures called Variable-tail Allocation Tables (VATs). A VAT, which is implemented as a linked list, is an array of pointers to the variable-length portion of Btrieve records. For this reason, VATs are enabled only when you have specified the Variable Records element.

Btrieve uses VATs not only to accelerate random access to the variable-length portions of records, but also to limit the size of the compression buffer used during data compression. If you have specified data compression, you may also want to specify VATs.

If you want to create a file that uses VATs, specify y. Otherwise, either specify n or omit this element from your description file.

For more information, see “Storing Variable-Length Records: Variable Tails and VATs” on page 209 in Appendix A.

Balanced Index

Format: `balance=<y|n>`

Range: Not applicable

Commands: CREATE

The Balanced Index element is optional. This feature allows you to use index balancing. With index balancing, Btrieve looks for available space in sibling index pages each time an index page becomes full and then rotates values from the full page into the pages with space available.

By default, Btrieve does not use index balancing. This means that each time a current index page becomes full, Btrieve must create a new index page. When you specify index balancing, Btrieve can often avoid creating a new index page each time a current index page is filled.

Specifying index balancing forces Btrieve to look for available space on adjoining index pages. If space is available on one of those pages, Btrieve moves keys from the full index page to the page with free space.

Index balancing increases index page utilization, results in fewer index pages, and produces more densely populated indexes. It also results in better overall disk usage and a faster response time on most read operations. In addition, index balancing produces an even distribution of keys among nodes on the same level, thus enhancing performance during get operations.

However, using index balancing also means that Btrieve requires extra time to examine more index pages in the file and may require more disk I/O during insert, update, and delete operations.

Although the exact effects of balancing indexes vary in different situations, performance on write operations typically degrades by about 5%-10% if you use index balancing.

Note

You can specify index balancing on a file-by-file basis when the file is created, or you can specify index balancing for all files. When you specify index balancing for a specific file, Btrieve will always balance that file's keys. You can also turn index balancing on and off for files that are not flagged as balanced by specifying Yes to the Perform Index Balancing configuration option in the Setup utility.

To use index balancing, specify y. Otherwise, either specify n or omit this element from the description file.

For more information, see "Index Balancing" on page 264 in Appendix A.

Key Position

Format: `position=`*`nnnn`*

Range: 1 through value specified for Record Length element (up to 4,088)

Commands: CREATE, INDEX, SINDEX

The Key Position element indicates the position of the key segment in the record.

The key position value must be at least 1 and cannot exceed the value you specified for the Record Length element.

For more information, see "Keys" on page 211 in Appendix A.

Key Length

Format: `length=`*`nnn`*

Range: 1 through limit determined by key type

Commands: CREATE, INDEX, SINDEX

The Key Length element defines the length of the key or key segment field.

The value you specify here cannot exceed the limit dictated by the key type, which the Key Type element (described on page 288) specifies.

The key length must be an even number if the key is a binary key type. The total of the key's length and starting position cannot exceed the file's defined record length.

For more information, see "Keys" on page 211 in Appendix A.

Duplicate Key Values

Format: `duplicates=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Duplicate Key Values element specifies whether more than one record in the file can contain the same value for this key field.

Specify y if you want to allow duplicate values for the key field; otherwise, specify n.

Note If you define duplicate key values for one segment of a segmented key, you must define duplicate key values for every segment of that key. For a segmented key that does not allow duplicates, the segments may contain duplicates between multiple records only if the key value is unique for each record.

For more information, see "Duplicatable and Nonduplicatable Keys" on page 214 in Appendix A.

Modifiable Key Values

Format: `modifiable=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Modifiable Key Values element specifies whether the key value can be modified during an update operation.

Specify y if you want the values for this key to be modifiable. Otherwise, specify n.

Note

If you define modifiable key values for one segment of a segmented key, you must define modifiable key values for every segment of that key because the key, as a whole, is modifiable.

For more information, see “Modifiable and Nonmodifiable Keys” on page 215 in Appendix A.

Key Type

Format: `type=`*`validBtrieveKeyType`*

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Key Type element specifies the Btrieve data type for the key. This element determines how Btrieve will sort the bytes specified for this key segment. Btrieve does not perform any validation on the data inserted.

You can specify the entire word (as in `float`) or just the first three letters of the word (as in `flo` for `float`). The keywords for the Btrieve key types are as follows:

autoinc

bfloat

date

decimal

float

integer

logical

lstring

money

numeric

string

time

unsigned binary

zstring

sign trailing separate

Note STS (`sign trailing separate`) is a COBOL data type. It is basically a numeric data type, represented as an ASCII string. STS is right justified and padded with leading ASCII zeros, and it has the sign byte at the end.

For more information, see “Key Types” on page 224 in Appendix A.

Descending Sort Order

Format: `descending=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Descending Sort Order element is optional. It specifies whether Btrieve will collate the index or index segment in descending order.

Btrieve normally orders key values in ascending order (lowest to highest). However, you can specify that Btrieve order the key values in descending order (highest to lowest) when you define the key segment.

Specify y if you want Btrieve to collate the key values in descending order. If you want Btrieve to collate the index in ascending order, either specify n or omit this element from the description file.

For more information, see “Descending and Ascending Keys” on page 216 in Appendix A.

Alternate Collating Sequence

Format: `alternate=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Alternate Collating Sequence element is optional. This element allows you to specify one or more alternate collating sequences (ACSs) for a given file. An alternate collating sequence (ACS) specifies whether Btrieve will sort a key by a collating sequence other than the standard ASCII sequence.

You can specify a different ACS for each key. However, each segment of the key needs to have the same ACS. ACSs are valid only for `string`, `lstring`, and `zstring` key types.

If you want the key to take advantage of an alternate collating sequence, specify y. Otherwise, either specify n or omit this element from the description file.

When using a description file to create an additional index for an existing Btrieve file, if you want the index to use an ACS file other than the first one in the Btrieve file, specify alternate=y.

If you specify `alternate=y` and `caseinsensitive=n` and use an ACS file other than the first one in the Btrieve file, the index will not be created.

For more information, see “Alternate Collating Sequence” on page 216 in Appendix A.

Case-Insensitive Key

Format: `caseinsensitive=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Case-Insensitive Key element is optional. It specifies that the key (or key segment) you are defining is case insensitive. You can specify this key element only for keys that are of type `string`, `zstring`, or `lstring`.

To specify that Btrieve ignore case when sorting the key value, specify y for this element. Otherwise, either specify n or omit this element from the description file.

For more information, see “Case-Insensitive and Case-Sensitive Keys” on page 216 in Appendix A.

Repeating Duplicates

Format: `repeatdup=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

Btrieve uses two methods for storing duplicatable keys internally. The keys are stored as either linked duplicatable keys or repeating duplicatable keys.

- Linked-duplicatable key—In a Btrieve v6.1 file, Btrieve stores duplicatable keys as linked-duplicatable keys by default on a create operation. (On a Create Index operation, Btrieve uses linked-duplicatable keys if they are available; otherwise, it uses repeating-duplicatable keys, as explained in the following section.) Using this method, Btrieve stores the key extracted from the first record of a duplicatable key on an index page.

  Other records with keys containing duplicate values are stored in the form of a linked list, with pointers at the end of each record in a data page pointing to the next and the previous records that have the same duplicate key values.

- Repeating-duplicatable key—If no room is available to create a linked-duplicatable key (that is, if no duplicate pointers were reserved at file creation, or if no index has been dropped to free existing pointers), Btrieve stores duplicatable keys as repeating-duplicatable keys. Using this method, Btrieve stores every key value of a repeating-duplicatable key both in a data page and in an index page.

Note Key-only files always use repeating-duplicatable keys because the key-only files use only the index pages and not the data pages.

Specify y to create repeating duplicatable keys. Otherwise, either specify n or omit this element from the description file.

Note For a segmented key, all segments must have the same `repeatdup=y/n` specification. For a nonsegmented key, if you specify `repeatdup=y`, you must also specify `duplicate=y`.

For more information, see “Linked-Duplicatable and Repeating-Duplicatable Keys” on page 214 in Appendix A.

Any Segment Null

Format: `nullkey=anyseg`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Any Segment Null element is optional. It specifies that the key or key segment you are defining contains a user-defined null character. An any segment null key does not include a particular record in the index if the value of any key segment of that record matches the null value.

Note

In previous releases, the Btrieve documentation referred to the Any Segment Null element at the Manual Key element.

Any segment null keys have all the attributes of a null key with one exception: in an any segment null key, Btrieve excludes from the index those records in which every byte in *any* segment of a key contains the null value.

If you define one segment of a key as an any segment null key, you must define a null value for that segment, and you must define all other segments of that key as any segment nulls. However, Btrieve allows you to define different null values for different segments in a segmented key.

To create an any segment null key or key segment, specify `anyseg`; otherwise, specify n or omit the `nullkey` keyword from your description file.

For more information, see “Null Keys (All-Segment and Any-Segment)” on page 222 in Appendix A.

All Segments Null

Format: `nullkey=allsegs|n`

Range: Not applicable

The All Segments Null element specifies whether the key you are defining has a null value. You can include the All Segment Null element in a description file for an indexing operation. However, to maintain consistent formats for the description files, the indexing operation disregards any null value you specify.

Note

In previous releases, the Btrieve documentation referred to the All Segments Null element as the Null Key element.

If you specify the All Segment Null key attribute, Btrieve excludes from the index those records in which *all* the bytes in *all segments* of a key contain a null value.

Specify `allsegs` if you want to define a null value for this key. Otherwise, specify `n` or omit the `nullkey` keyword.

For more information, see "Null Keys (All-Segment and Any-Segment)" on page 222 in Appendix A.

Null Key Value

Format: `value=`*`nn`*

Range: Any 1-byte hexadecimal value

Commands: CREATE, INDEX, SINDEX

The Null Key Value element specifies the null character value (in hexadecimal) for the key. Typical null values are 20 hexadecimal (blank) and 0 hexadecimal (binary zero).

Include this element only if you defined the key as allowing null values.

If you specify `nullkey=n`, the description file does not require the Null Key Value element.

For more information, see "Null Keys (All-Segment and Any-Segment)" on page 222 in Appendix A.

Segmented Key

Format: `segment=<y|n>`

Range: Not applicable

Commands: CREATE, INDEX, SINDEX

The Segmented Key element specifies whether the key you are defining has any more segments.

When you assign an attribute to a key, all segments of that key share the same attributes. However, the key type can be different for each segment in the key. The sort order, either ascending or descending, can also differ for each segment.

A Btrieve file limits the maximum number of *key segments* rather than the maximum number of keys. The maximum number of key segments allowed depends on the page size. (Note that you can define more key segments than there are keys defined for the file.)

Table 22 on page 280 shows the maximum number of key segments, *not* the number of keys, for each possible page size.

Note If you define a key as duplicatable, Btrieve allows more than one record to have the same value for that key. If you specify a key as nonduplicatable, Btrieve does not allow an application to add multiple records with the same key value. If one segment of a key allows duplicates, all segments must allow duplicates.

Specify y if the key has another segment. Specify n if you are defining either a nonsegmented key or the last segment of a segmented key.

For more information, see “Segmented and Nonsegmented Keys” on page 212 in Appendix A.

Alternate Collating Sequence Filename/ID

Format: `name=`*`sequenceFile`* or `countryid=`*`nnn`* `codepageid=`*`nnn`*

Range: Any valid, fully qualified pathname; or a valid country ID and code page ID; or –1 (the default)

Commands: CREATE, INDEX, SINDEX

The Alternate Collating Sequence Filename/ID element specifies the pathname of the file that contains an alternate collating sequence for the file you are creating. You can include up to 256 bytes of directory levels in the pathname (plus the filename).

If you specified n for the Alternate Collating Sequence element for every key, do not include this element in your description file.

If you specified y for the Alternate Collating Sequence element for a key, you must supply either an alternate collating sequence filename *or* a country ID and a code page ID.

If you want all the keys in the file to use the same ACS, you can either specify this element as the last element in the description file or specify the alternate collating sequence name for each key.

If you want to use different ACSs, you must specify this element for each key that uses an alternate collating sequence.

You can specify only one ACS per key, and each segment of the key should have an `alternate=` and `name=` pair, or a `countryid=` and `codepage=` pair.

To use an ACS ID, you must specify `countryid=`*`nnn`* and `codepageid=`*`nnn`*. Use a valid country ID and code page ID. If you want to use the current locale, specify `countryid=–1` and `codepageid=–1`.

The first 265 bytes of an ACS file contain the definition of a collating sequence other than the standard ASCII sequence. If you want to create an ACS file, generate a file in the format that Table 23 specifies.

Table 23

Alternate Collating Sequence File Format

Offset	Length	Description
0	1	Signature byte. This byte should always contain the value ACh.
1	8	An 8-byte name that uniquely identifies the alternate collating sequence to Btrieve.
9	256	A 256-byte table containing the sort value for every character. Store the value for each sort character at the offset corresponding to the character's representation in the ASCII collating sequence. For example, to sort the character *A* as something other than 41h, store the new sort value at offset 41h in the table.

For example, assume you want to insert a character with a 5Dh between the letters U (55h) and V (56h) in your sequence. In this case, byte 5Dh in the sequence should contain the value 56h, and bytes 56h through 5Ch in the sequence should contain the values 57h through 5Dh.

For more information, see "Alternate Collating Sequence" on page 216 in Appendix A.

appendix C Status Codes and Messages

This appendix describes the Btrieve® status codes and messages. The codes are listed first and appear in numeric order. The messages follow the codes and are listed in alphabetic order.

Status Codes

This section lists and describes the status codes that the NetWare Btrieve 6.*x* Record Manager, client-based Btrieve for OS/2 and MS Windows, and the Btrieve Requesters can return. Table 24 shows the numeric ranges dedicated to each type of code.

Table 24
Status Code Ranges

Range	Type of Code
0 – 199	NetWare Btrieve Record Manager Status Codes
1000–1999	Client-Based Btrieve for OS/2 and MS Windows Status Codes
2000–2099	Btrieve Requester Status Codes

NetWare Btrieve Record Manager Status Codes

The NetWare Btrieve Record Manager returns a status code after each operation that an application performs. If the operation is successful, Btrieve returns Status Code 0.

If the operation is not successful, Btrieve returns one of the nonzero status codes described in this section.

01: The operation parameter is invalid.

The operation parameter specified in the call is invalid.

02: Btrieve encountered an I/O error.

Btrieve encountered an error while reading from or writing to the disk. One of the following has occurred:

- The file is damaged and must be recreated.
- There is a large pre-image file inside a transaction, and there is not enough space for a write to the pre-image file.

Note

Pre-image files are used only for files created by Btrieve versions earlier than 6.*x*, *or* by 6.*x* if it was loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option set to Yes.

- For Btrieve 5.*x* files, there is one pre-image file for multiple data files. For example, if you name the data files CUSTOMER.ONE and CUSTOMER.TWO, both files will have pre-image files named CUSTOMER.PRE.
- There is not enough space to append a new page to the data file.
- Client-based Btrieve attempted to write to a Btrieve file flagged shareable while server-based Btrieve had the file open.

03: The file is not open.

The operation cannot be executed because the file is not open. The application must perform a successful Open operation before Btrieve can process any other operations.

This status code may also be returned if the application passed an invalid position block for the file, or if the application passed a position block with a different client ID than the client ID used to open the file.

04: Btrieve cannot find the key value.

Btrieve cannot find the specified key value in the index path.

Also, pre v6.*x* versions of NetWare Btrieve could return this status code if two separate files have different alternate collating sequences (ACSs), but those ACSs have the same name. Never use the same for different ACSs, regardless of the version of Btrieve you are using.

05: The record has a key field containing a duplicate key value.

Btrieve cannot add or update a record for the designated index because the record has a key field that contains a duplicate key value, and the index does not allow duplicate values.

06: The key number parameter is invalid.

The value stored in the key number parameter is not valid for the file being accessed. The key number must correspond to one of the keys defined for the file. Valid key numbers are 0 through 118.

07: The key number has changed.

The key number parameter changed before a Get Next, Get Next Extended, Get Previous, Get Previous Extended, Update, or Delete operation. The operation requires the same key number parameter as the previous operation because Btrieve uses positioning information relative to the previous key number.

Another related situation can also result in Btrieve returning this status code. That situation occurs when an application performs a Delete or Update operation immediately following a Get operation.

If the application changes the value of the key number in the Delete or Update operation (from the value returned by the preceding Get operation), Btrieve will delete or update the record as requested and will *not* return Status Code 07, at least not at this point. However, Btrieve *will* return Status Code 07 on the very first Get operation performed after the deletion or update, even if that Get operation uses the same key value the application passed to the Delete or Update operation.

If you need to change key numbers between consecutive Get Next, Get Next Extended, Get Previous, Get Previous Extended, Update, or Delete operations (or in the Delete or Update operations as described in the preceding paragraphs), use a Get Position operation followed by a Get Direct operation to reestablish positioning for the new index path.

08: The current positioning is invalid.

The current position must be established to update or delete a record. Perform a Get or Step operation to establish the current position.

This status code may also be returned if the application passed an invalid position block for the file.

09: The operation encountered an end-of-file condition.

This status code indicates one of the following conditions has occurred:

- The operation encountered an end-of-file boundary or tried to read past a file boundary (end-of-file or start-of-file).
- In a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, the number of records satisfying the filtering condition is less than the number of specified records to be returned, and the reject count has not been reached.
- When reading a file in ascending order according to an index path, Btrieve has already returned the last record in that index path. When reading a file in descending order according to an index path, Btrieve has already returned the first record in the index path.
- When using the Get By Percentage operation, either the value supplied for the percentage is too high—that is, it exceeds 10,000 decimal (0x2710)—or the file contains no records.

10: The key field is not modifiable.

During an Update operation, an attempt was made to modify a key field that is defined as nonmodifiable.

11: The specified filename is invalid.

This status code indicates one of the following conditions is true:

- The filename specified does not conform to file naming conventions. Make sure the filename is valid for the environment.
- An attempt was made to open a file that has .^^^ as its extension. This extension is reserved for Btrieve to use during a continuous operation.
- The data buffer for a Begin or End continuous operation is not set up correctly.

If you receive this status code when you open a Btrieve file using the NetWare Btrieve OS/2 Requester (BTRCALLS.DLL), make sure that you are using the latest NetWare OS/2 Requester.

12: Btrieve cannot find the specified file.

Check the key buffer parameter to make sure the pathname is terminated with a blank or a binary zero. Also, check to be sure the file exists.

When accessing a Btrieve file on a NetWare file server, be sure that you have FILE SCAN rights to the directory in which the Btrieve file resides.

14: Btrieve cannot create or open the pre-image file.

There are four possible causes for this status code:

Note Pre-image files are used only for files created by Btrieve versions earlier than 6.*x*, *or* by 6.*x* if it was loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option set to Yes.

- Btrieve cannot create a new pre-image file because the disk directory is full. Btrieve must be able to create a pre-image file.

- Btrieve cannot open the pre-image file to restore file integrity. If the pre-image file is erased or damaged, Btrieve cannot restore the file's integrity. In this case, either use the RECOVER command in the Btrieve Maintenance utility to retrieve the damaged file's data records in a sequential file, or replace the file with its most recent backup.

- Client-based Btrieve cannot assign a handle to the pre-image file because Btrieve was not started by a user with access rights to the pre-image file.

- The file structure of a pre-image file created by Btrieve 6.*x* is different from the file structure of a pre-image file created by Btrieve 5.*x*. If you have an extraneous .PRE file in Btrieve 5.*x* format and you are using Btrieve 6.*x*, Status Code 14 is returned when you try to open the Btrieve file to which the .PRE file belongs.

15: Btrieve encountered an I/O error during pre-imaging.

This status code indicates either the disk is full or the pre-image file is damaged.

Note Pre-image files are used only for files created by Btrieve versions earlier than 6.*x*, *or* by 6.*x* if it was loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option set to Yes.

When this status code occurs, proceed as follows:

- If the disk is full, erase any unnecessary files or extend the file to gain additional disk space.
- If the pre-image file is damaged, the integrity of the Btrieve file cannot be ensured. Either use the RECOVER command in the Btrieve Maintenance utility to retrieve the damaged file's data records in a sequential file, or replace the file with its most recent backup.

16: Btrieve encountered an expansion error.

Pre-6.0 versions of Btrieve return this status code when they encounter an error while writing the directory structure to disk prior to creating the expanded file partition. Either Btrieve cannot close the file, or a new page was added to the file and Btrieve cannot close and reopen the file to update the directory structure. Check for a disk hardware failure.

17: Btrieve encountered a close error.

Btrieve encountered an error while writing the directory structure to disk prior to closing the file. Either Btrieve cannot close the file, or a new page was added to the file and Btrieve cannot close and reopen the file to update the directory structure. Check for a disk hardware failure.

This status code may also be returned if the application passed an invalid position block for the file.

18: The disk is full.

This status code can be returned in the following situations:

- The disk is full, and the file cannot be expanded to accommodate additional records. Either erase any unnecessary files or extend the file to gain additional disk space. If using a version of Btrieve prior to 6.0, you can possibly extend the file to gain additional disk space.
- The pre-image file is out of disk space. If you are working with files created by Btrieve versions earlier than 6.*x* and you are in a transaction, the pre-image file size increases for the duration of the transaction. If you receive this status, either reduce the number of operations in the transaction or obtain more disk space.

- The NetWare® owner name for the file is no longer valid, and your application tried to insert or update records in the file, thus causing the file to expand. In this case, this status code is returned when Btrieve needs to add a page to the file, regardless of how much disk space is available. To check for an owner name, use the NetWare utility NDIR. To add an owner name, use FILER (a NetWare text utility) or the NetWare Administrator graphical utility.

- You can limit the amount of disk space available to each user in NetWare 4. This status code indicates an attempt was made to expand a Btrieve file beyond the amount of disk space allocated to the file's owner in NetWare.

19: The application encountered an unrecoverable error.

To ensure file integrity, either use the RECOVER command in the Btrieve Maintenance utility to retrieve the damaged file's data records in a sequential file, or replace the Btrieve file with its most recent backup.

20: The Record Manager or Requester is inactive.

You must load Btrieve and, if applicable, the Btrieve Requester before generating any requests. Also, in the MS Windows environment, be sure that the NetWare Communication Services DLLs and WBTR32.EXE are in your PATH or in the top level of your MS Windows directory

21: The key buffer parameter is too short.

The key buffer parameter is not long enough to accommodate the key field for the index path requested. Verify that the length of the key buffer equals the defined length of the key specified in the key number parameter. Only language interfaces that track the buffer length can return this status code.

22: The data buffer parameter is too short.

The data buffer parameter is not large enough to accommodate the length of the data record defined when the file was created. Verify that the length of the data buffer is at least as long as the file's defined record length.

- For Get or Step operations, Btrieve returns as much data as it can and a Status Code 22, indicating that it cannot return the entire record.

- For an Insert operation, Btrieve does not insert the record if the data buffer is shorter than the fixed-length portion of the record.

- For an Update operation, if the data buffer is too short to contain the fixed-length portion of a record, Btrieve does not update the record.

- For the Create, Stat, and Create Supplemental Index operations, the data buffer is not long enough to contain all the file specifications, the key specifications, and (if specified) the alternate collating sequence definition.

- For the Get By Percentage or Find Percentage operation, the data buffer length is less than 4 bytes.

23: The position block parameter is not 128 bytes long.

Only language interfaces that track the position block length can detect and return this status code. None of the currently shipping language interfaces return Status Code 23.

24: The page size or data buffer size is invalid.

Two possible reasons for receiving this status code are as follows:

- The page size is invalid. The page size must be a multiple of 512 bytes and cannot exceed 4,096 bytes.

- During a Create operation, the page size is the first file specification Btrieve checks. A Status Code 24 at this point may indicate an invalid data buffer parameter.

In versions prior to Btrieve 6.1, this status code can be returned from the Open operation. In this case, Btrieve cannot open the file because the file's page size exceeds the Largest Page Size configuration option. To successfully open the file, you must increase the value of the Largest Page Size configuration option and then reload Btrieve.

Btrieve 6.1*x* will not return Status Code 24 from the Open operation.

25: Btrieve cannot create the specified file.

Btrieve can return this status code if an application attempts to create a Btrieve file, but the disk directory or the disk itself is full. If the application is creating a file over an existing file, Btrieve returns this status code when the existing file is open or when the operating system prevents the operation for some other reason (for example, because the file is flagged transactional).

Some pre-6.*x* versions of Btrieve can also return Status Code 25 if the HOLD parameter in NET.CFG or SHELL.CFG is set to ON, and the application attempts to create a Btrieve file on a network drive. (The HOLD parameter is set to OFF by default.) Creation of the Btrieve file on a local drive will be successful regardless of the value of the HOLD parameter.

26: The number of keys specified is invalid.

The number of keys specified for the page size is invalid. The number of key segments must be within the following limits:

Page Size	**Max. Number Key Segments**
512	8
1,024	23
1,536	24
2,048	54
2,560	54
3,072	54
3,584	54
4,096	119

27: The key position is invalid.

The key field position specified is less than 1 or exceeds the defined record length for the file. Either the key position is greater than the record length or the key position plus the key length exceeds the record length.

28: The record length is invalid.

The record length specified (plus overhead for duplicates, record usage count, variable record pointers, record length, and blank truncation information) must be less than or equal to the page size minus 8 bytes, and greater than or equal to 4 bytes.

29: The key length is invalid.

The key length specified must be greater than 0 but cannot exceed 255 bytes. The length of a binary key must be an even number. Btrieve requires that each key page in the file be large enough to hold at least eight keys.

If the page size is too small to accommodate eight occurrences of the specified key length (plus overhead), either increase the file's page size or decrease the key length.

30: The file specified is not a Btrieve file.

Either Btrieve did not create the file, or a version of Btrieve earlier than 3.*x* created it.

NetWare Btrieve 6.0 returns this status code when you attempt to open a file that has the NetWare Btrieve 6.1*x* file format. (This is generally true whenever you use an earlier NetWare Btrieve version to open a file that has a more recent file format.)

This status code can also indicate that the first page of the file is damaged. Use a backup copy of your data file.

31: The file is already extended.

This status code is returned by pre-6.0 client- and VAP-based versions of Btrieve if an application tried to specify a file that has already been extended. A file can be extended only once.

Files on a NetWare 3 or NetWare 4 server using the NetWare Btrieve NLM cannot be extended.

32: The file cannot be extended.

This status code is returned by pre-6.0 client- and VAP-based versions of Btrieve if an application tried to specify a file that cannot be extended. Possible causes for receiving this status code are that the directory is full, the disk is full, or the disk is write protected.

34: The specified extension name is invalid.

This status code is returned by pre-6.0 client- and VAP-based versions of Btrieve if an application specified an invalid filename for the extended partition. Check the validity of the filename.

35: Btrieve encountered a directory error.

Either a Get Directory operation specified a drive that does not exist, or a Set Directory operation specified an invalid pathname. Check the validity of both the drive and the pathname.

36: Btrieve encountered a transaction error.

Btrieve tried to perform a Begin Transaction operation without configuring Btrieve to allow transactions. Run the Setup utility and specify a higher value for the Number of Transactions configuration option. Next, stop and then restart Btrieve using BSTOP and BSTART so that your changes will take effect.

On a workstation that is running both client-based Btrieve and the Btrieve Requester, be sure that both the client engine and the Requester are configured for transactions.

37: Another transaction is active.

The application issued a Begin Transaction operation while another transaction was active by the same client (it can be an NLM or application). Btrieve does not accommodate nesting transactions.

38: Btrieve encountered a transaction control file I/O error.

Btrieve encountered an error when it tried to write to the transaction control file. Possible causes for receiving this status code are that the disk is full, the disk is write protected, the transaction control file (BTRIEVE.TRN) that is created when you load Btrieve has been deleted or the transaction control file is flagged read only.

39: A Begin Transaction operation must precede an End/Abort Transaction operation.

The application issued an End or Abort Transaction operation without a corresponding Begin Transaction operation. Make sure that each End or Abort Transaction operation in your program has a corresponding Begin Transaction operation.

40: The file access request exceeds the maximum number of files allowed.

The application tried to access more than the maximum number of files allowed within a transaction. The maximum number of different files that can be accessed during a logical transaction is set when Btrieve is configured.

Note: This status code applies only to Btrieve versions earlier than 6.*x*.

41: Btrieve does not allow the attempted operation.

This status code is returned for one of the following reasons:

- The application tried to perform an operation that is not allowed at this time. Btrieve does not allow some operations under certain operating conditions. For example, Btrieve returns this status code if the application attempts to perform a Step operation on a key-only file or a Get operation on a data-only file.

- The key number parameter of a continuous operation call is not 0, 1, or 2.

Also, Btrieve prohibits certain operations during transactions because they have too great an effect on the file or on Btrieve's performance. These operations include Set Owner, Clear Owner, Extend, Create Supplemental Index, and Drop Supplemental Index.

42: A file previously opened in Accelerated mode was not closed.

Either the application tried to open a Btrieve 5.*x* file that was previously accessed in Accelerated mode by Btrieve 5.*x* and never successfully closed, or the application tried to open a file for which Btrieve 6.*x* encountered an unrecoverable error during a Set or Clear Owner operation. The file's integrity cannot be ensured. Either use the RECOVER command in the Btrieve Maintenance utility to retrieve the damaged file's data records in a sequential file, or replace the file with its most recent backup.

Another possible cause for receiving Status Code 42 is that your application tried to open a 5.*x*-formatted Btrieve file using a 5.*x* Btrieve engine. However, that same file was previously accessed by a Btrieve 6.*x* engine, which failed to close the file successfully due to a crash. Btrieve 5.*x* does not understand the format of a pre-image file created by Btrieve 6.*x*.

43: The specified record address is invalid.

This status code is returned for one of the following reasons:

- The record address specified for a Get Direct operation is invalid. The address is outside the file's boundaries, it is not on a record boundary within a data page or on a data page, or the record at the specified address has been deleted. For a Get Direct operation, specify the 4-byte address obtained by a Get Position operation.

- When using Btrieve with NetWare and the files are Btrieve 5.*x* files, this error may indicate a file access conflict. For example, workstation 1 has a file locked in an exclusive transaction. Workstation 2 is reading records from the same file and tries to update a record that the transaction either inserted or updated. If workstation 2 reads the record and then workstation 1 aborts the transaction, workstation 2 receives this status code when issuing the Update operation.

- For a Find Percentage operation that is seeking a percentage based on a record's physical location within the file, the specified record address is invalid.
- The file may be corrupt, and you must rebuild it.

The Btrieve Roll Forward utility receives this status code when the address acquired from the Free Space List of the Btrieve file during an Insert operation at roll forward time is different from the one received at logging time.

The synchronization problem with Free Space Lists occurs when you use different versions of Btrieve or the same Btrieve version but with a different index balancing setting.

The following is a summary of the specific conditions for which Status Code 43 can be returned to the Roll Forward utility:

- During logging, you loaded Btrieve multiple times and did not specify the same index balancing setting each time.
- You inserted or updated a record larger than 57 KB.
- At roll forward time, you loaded a different version of NetWare Btrieve or the same NetWare Btrieve version but with a different index balancing setting from what you used at logging time.To correct the synchronization problem with Free Space Lists, load the proper version of NetWare Btrieve with the proper index balancing setting.

44: The specified key path is invalid.

The application tried to use the Get Direct operation to establish an index path for a key whose value is null in the corresponding record. Btrieve cannot establish positioning based on a null key value.

NetWare Btrieve 6.1*x* returns this code for some situations in which earlier versions of NetWare Btrieve returned Status Code 82 (The application lost positioning). These situations can occur when you define the key attribute to be manual or null (all-segment and any-segment).

If you have an existing NetWare Btrieve application that checked for Status Code 82, you may want to revise the application to check for Status Code 44 as well.

45: The specified key flags are invalid.

The key flags specification on a Create operation is inconsistent. If a key has multiple segments, the duplicate, modifiable, and null attributes should be the same for each segment in the key. Also, you cannot use the Null or Manual key attributes in a key-only file.

This status code is also returned if an attempt is made to specify a different alternate collating sequence for two or more segments of a segmented key.

46: Access to the requested file is denied.

This status code is returned for one of the following reasons:

- The application opened a file in read-only mode and tried to perform an Insert, Update, or Delete on that file.
- An attempt was made to perform an Insert, Update, or Delete on a file that is flagged read-only to NetWare.
- The owner name required for updates was not specified correctly when the file was opened.

47: The number of files opened exceeds the maximum allowed.

The number of files opened in Accelerated mode exceeded the number of buffers available in Btrieve’s cache. When a file is opened in Accelerated mode, Btrieve reserves one of its cache buffers for the file. Btrieve always reserves five empty buffers for index manipulation.

In client-based Btrieve, you should reconfigure Btrieve with a smaller /P configuration option to allocate more buffers and a larger /M option. In server-based Btrieve, you will not encounter this error code.

48: The alternate collating sequence definition is invalid.

The first byte of an alternate collating sequence definition (the identification byte) does not contain the hexadecimal value AC (for user-defined ACSs) or AD (for locale-specific ACSs). Make sure that the first byte contains the proper value.

49: The extended key type is invalid.

You tried to create a file or a supplemental index with an invalid extended key type, or you tried to assign an alternate collating sequence to a binary key or key segment. You can assign an alternate collating sequence only to a `string`, `lstring`, or `zstring` key type.

This status code is also returned if you define a supplemental index requiring an alternate collating sequence, but no alternate collating sequence definition exists (either in the file or in the key definition passed in the data buffer).

Another possibility is that you defined an alternate collating sequence with case insensitivity. These two definitions are incompatible. Only Btrieve 6.0 interprets this condition as an error.

A final possibility is that you are attempting to create a Btrieve file that contains multiple alternate collating sequences but your server has a version of Btrieve loaded that predates 6.1.

50: The file owner is already set.

The application tried to perform a Set Owner operation on a file that already has an owner. Use the Clear Owner operation to remove the previous owner before specifying a new one.

51: The owner name is invalid.

The possible causes for this status code are as follows:

- If the application received this status code after a Set Owner operation, the owner names specified in the key buffer and data buffer do not match.

- If this status code occurred during an Open operation, the application attempted to open a file that has an owner name assigned to it. The application must specify the correct owner name in the data buffer.

- If an NLM received this status code when dealing with a file in continuous operation mode, then the client ID of the calling NLM differs from the client ID of the application that originally put the file into continuous operation mode.

53: The language interface version is invalid.

An application tried to access a file containing variable-length records with a language interface from Btrieve 3.15 or earlier. To access files with variable-length records, you must use a 4.*x* or later interface.

54: The variable-length portion of the record is corrupt.

During a Get or Step operation, Btrieve cannot read all or part of the variable-length portion of a record. Btrieve returns as much data as possible to the application. This status code usually indicates one or more pages used to store variable length records is corrupt. Use BUTIL –RECOVER to recover as much data as possible.

55: The application specified an invalid attribute for an autoincrement key.

The application tried to specify either the segmented or duplicate attribute for an `autoincrement` key type. An `autoincrement` key cannot be part of a segmented key and cannot allow duplicates.

56: An index is incomplete.

An index can be damaged if a Create Index operation (31) or a Drop Index operation (32) is interrupted before it runs to completion. Perform a Drop Index operation to completely remove the damaged index from the file, and then rebuild the index with the Create Index operation, if so desired.

58: The compression buffer length is too short.

The application tried to read or write a record that is longer than the value specified for the size of the compression buffer. Reconfigure Btrieve using the Setup utility, specifying a higher value for the Largest Compressed Record Size option.

59: The specified file already exists.

This status code is returned for the Create operation if the application specified –1 in the key number parameter and the name of an existing file in the key buffer parameter. If you want to overwrite the existing file, remove the –1 from the key number parameter. If you want to preserve the existing file, alter the filename specified in the key buffer parameter.

60: The specified reject count has been reached.

Btrieve rejected the number of records specified by the reject count before a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation found the requested number of records that satisfy the filtering condition. Check the first two bytes returned in the data buffer for the number of records that were retrieved.

61: The work space is too small.

The Get Next Extended, Get Previous Extended, Step Next Extended, and Step Previous Extended operations use a buffer as work space. This status code indicates that the work space (set by default to 16 KB) is not large enough to hold the filtering data buffer structure and the largest record to be received.

62: The descriptor is incorrect.

The descriptor (data buffer structure), which is passed for a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, is incorrect. The descriptor length (the first two bytes of the data buffer) on the extended operation call must be the exact length of the descriptor. This requirement does not apply to the data buffer length option, which can still be declared longer than necessary.

On a Get Direct/Chunk operation, the descriptor structure in the data buffer is incorrect, or it is inconsistent (either internally or in respect to the data buffer length).

63: The data buffer parameter specified on an Insert Extended operation is invalid.

An Insert Extended operation provided an invalid buffer. Either the buffer length is less than 5 bytes, or the number of records specified is 0. Correct the buffer length or the number of records.

64: The filter limit has been reached.

During a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation, a rejected record was reached; no other record can satisfy the given filtering condition, going in the direction that the operation specified. This is applicable only if the first segment of the key that the key number specified is also used as the first term of the filtering field.

65: The field offset is incorrect.

The field offset in the extractor of a Get Next Extended, Get Previous Extended, Step Next Extended, or Step Previous Extended operation is invalid based on the length of the retrieved record. Make sure that the field offset is a valid value (from 0 through the record length minus 1).

66: The maximum number of open databases has been exceeded.

The application has tried to open too many SQL databases configured for referential integrity checking at one time. The maximum number of databases that can be opened concurrently for referential integrity checking is a configurable startup option for Btrieve. The default is set to 20.

Refer to your Scalable SQL documentation for more information on referential integrity.

67: Btrieve cannot open the SQL data dictionaries.

This status code may indicate that an application opened a data file containing referential integrity definitions but Btrieve cannot open one of the Scalable SQL data dictionary files (FILE.DDF or RELATE.DDF) or the configuration file (DBNAMES.CFG).

If you are running NetWare Btrieve, be sure that DBNAMES.CFG is in the SYS:SYSTEM directory on the server running the Scalable SQL NLM.

If you are running a client-based version of Btrieve, be sure that DBNAMES.CFG is in your Novell database directory on the local drive; or if accessing Scalable SQL remotely, be sure that DBNAMES.CFG is in the SYS:SYSTEM directory on the server running the Scalable SQL NLM.

Regardless of which Btrieve version you are running, be sure that the FILE.DDF and RELATE.DDF files for the named database are placed in the dictionary location that the NetWare SQL Setup utility defines.

Refer to your Scalable SQL documentation for more information on referential integrity.

68: Btrieve cannot perform the RI Delete Cascade operation.

Btrieve cannot enforce the Delete Cascade rule on a file under referential integrity control because the record that the application attempted to delete has more than 16 levels of descendants. Delete records from the lower levels, and then try again to delete the record that the application was attempting to delete initially. Refer to your Scalable SQL documentation for more information on referential integrity.

69: The Delete operation cascades to a record in a file that is damaged.

The application encountered an error while Btrieve was attempting to enforce the Delete Cascade rule in response to a Delete operation. This status code indicates that the related file has been damaged and must be recreated.

Refer to your Scalable SQL documentation for more information on referential integrity and the Delete Cascade rule.

71: There is a violation of the RI definitions.

If you have attempted an Insert operation on a file under referential integrity control, you may receive this status code if a foreign key value in the record to be inserted does not have a corresponding primary key in the referenced file.

If you are performing an Update operation, there are two possible causes for this status code:

- You are attempting to change the value of a primary key.
- You are attempting to change the value of a foreign key to a value that does not exist for the defined primary key.

If you have attempted a Delete operation, the Restrict rule is being enforced, and a primary key value in the record you are trying to delete references a foreign key in the referenced file. Refer to your Scalable SQL documentation for more information on referential integrity.

72: Btrieve cannot open the RI referenced file opened.

The referenced file cannot be found at the location that FILE.DDF and DBNAMES.CFG specify. This status code is returned only on a first attempt to delete or insert in a file under referential integrity control, or on a first attempt to update when that operation would change a foreign key in the specified file.

If running NetWare Btrieve, verify that the file location does not contain a drive letter.

If running client-based Btrieve *and* if using drive letters, make sure that the drive letters are the same (and map to the same locations) as specified in DBNAMES.CFG.

Refer to your Scalable SQL documentation for more information on referential integrity.

73: The RI definition is out of sync.

You have attempted to open a file for which the referential integrity definition either disagrees with the definition in RELATE.DDF or refers to a database not found in DBNAMES.CFG. This status code can also occur on an Insert or Delete operation, or an Update operation that would change a foreign key.

If your Scalable SQL is version 3.0, run the RIUTIL utility and use the **`check`** command. If you have a later version of Scalable SQL, run the SQLScope utility and use the **`Check Constraints`** command under the Database menu. For information on these utilities and on referential integrity, refer to your Scalable SQL documentation.

74: Btrieve aborted the transaction.

This is an informative status code. A pre-6.*x* version of NetWare Btrieve replaced an End Transaction operation with an Abort Transaction after detecting an error for a Transaction Tracking System (TTS) file inside the transaction. In addition, Btrieve executed the Abort Transaction operation.

Status Code 74 will not be returned by 6.*x* NetWare Btrieve or by client-based Btrieve because TTS has a different meaning to 6.*x* NetWare Btrieve, and client-based Btrieve ignores the TTS flag.

76: There is a conflict on the referenced file.

The application has attempted to perform an Update, Insert, or Delete operation on an RI-controlled file that references another file. The application cannot open the referenced file for referential integrity checking because it is already opened in Exclusive mode. Wait until the referenced file is closed or is opened in a mode other than Exclusive, and then retry the operation. Refer to your Scalable SQL documentation for more information on referential integrity.

77: The application encountered a wait error.

NetWare Btrieve returns this status code in one of the following situations:

- A wait lock bias is specified for an operation, but another user has locked the requested resource.
- The application is currently processing a wait transaction and tried to access a file that another user has locked.

When you are using the Btrieve Requester to access Btrieve, the Requester waits and retries if a requested resource is locked. When a server-based application, such as Scalable SQL, is accessing Btrieve and the requested resource is locked, a wait is also required. In this case, Btrieve would be expected to perform the wait. Since this would occupy Btrieve and lock out other users who might be trying to release the requested resource, Btrieve does not perform the wait. Instead, it returns a Status Code 77, and the server-based application must retry later.

78: Btrieve detected a deadlock condition.

The application should clear all resources (for example, by aborting or ending the transaction, or releasing all record locks) before proceeding. This breaks the deadlock, allowing other applications to access the resources for which they are waiting.

79: A programming error occurred.

Although very rare, it is possible to receive this status code when there is a malfunction that Btrieve cannot specifically detect or from which Btrieve cannot recover. Retry the operation again. If the error persists, there may be a system corruption; try to clear the system by rebooting, and then try the operation again.

80: The application encountered a record-level conflict.

Btrieve did not perform the Update or Delete operation because of a record-level conflict. For example, station A reads a record, station B reads the same record and updates it, and then station A attempts to update the record. The application should reread the record prior to resending an Update or Delete operation.

81: Btrieve encountered a lock error.

This status code can result from one of the following conditions:

- The Btrieve lock table is full. Decrease the number of locks that the application uses, or use the Setup utility to specify a higher value for the Number of Locks option.

- The application tried to unlock one record that is locked with a multiple record lock, but the record position stored in the data buffer does not correspond to any record locked in the associated file.

- The application tried to unlock a single-record lock with a multiple-record lock or vice-versa.

82: The application lost positioning.

When performing a Get Next or Get Previous operation on a key with duplicates, the application tried to retrieve a record that was deleted or whose key value was modified by another application. Use a Get Equal or a Get Direct operation to reestablish positioning. (See Status Code 44 for a related positioning problem.)

83: The application attempted to change a record from within a transaction; however, the record was read outside the transaction.

The application tried to update or delete a record within a transaction, but the record was not read within the transaction. The application must read the record within the transaction before attempting to modify the data.

84: The record or page is locked.

The application tried to apply a nowait lock on a record that is currently locked by another application, or the application tried to access a file in a nowait transaction while another application holds an active record lock(s) in that file.

This status code can also occur if the application tried to update or delete a record locked by another application.

The application can use either of the following recovery methods:

- Retry the operation until it is successful. Under light-to- moderate network use, this may be the simplest and quickest solution.
- Use the wait option (+100/+300) instead of the nowait option.

Btrieve may return this status code on an Insert operation when it attempts to lock an index page to update the key value specified for a newly inserted record. If another user has already locked the record in question, then Btrieve returns a Status Code 84 when it attempts to update the key value. Have your application check for this status code and retry the operation if the status code is returned.

85: The file is locked.

Any of the following can cause this status code to occur:

- Client-based Btrieve has a file open, and another workstation that has the Requester loaded tries to open the same file. Btrieve cannot open the file since it cannot obtain exclusive access. The workstation that has the Requester loaded receives Status Code 85.

- Another workstation has the Requester loaded and has a file open, and client-based Btrieve tries to open the same file.

- A file is in transition into continuous operation mode. A retry will eventually work.

- You have two Btrieve files with the same filename but different extensions (for example, INVOICE.HDR and INVOICE.DET). One file is open and in continuous operation mode, causing Btrieve to generate a delta file (for example, INVOICE.^^^). Btrieve returns Status Code 85 if you attempt to open the second file.

- While one user has a file locked in an exclusive transaction, another user attempts to lock all or part of that file.

86: The file table is full.

Using the Setup utility, specify a higher value for the Number of Open Files configuration option. For more information, refer to "Number of Open Files" on page 64 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

87: The handle table is full.

This status code is applicable only in the server-based Btrieve environment. Using the Setup utility, specify a higher value for the Number of Handles configuration option. For more information, refer to "Number of Handles" on page 64 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

88: The application encountered an incompatible mode error.

This status code can indicate one of the following situations:

- If an application opens a file in Exclusive mode, all other applications get this status code when they try to open the same file in any mode.

- If an application opens a file in any mode other than Exclusive, all other applications get this status code when they try to open the same file in Exclusive mode.

In NetWare Btrieve's continuous operation mode, this status code can also indicate one of the following situations:

- You have attempted to remove a file from continuous operation, but the file is not in a continuous state.
- You have attempted to include two files that have the same name but different extensions in continuous operation.
- You have attempted to include a file in continuous operation, but the file is already in a continuous state.

91: The application encountered a server error.

You can receive this status code in the following situations:

- The Requester cannot establish a session with the server. Either Btrieve is not loaded or the server is not active.
- The setting for the Number of Remote Sessions configuration option is too low. Return to the Setup utility and specify a higher value for this option.
- An application specified a path for a file and did not include the volume name in the path.
- The Btrieve Message Router (BROUTER.NLM) has not been loaded, and the following situation has occurred: an NLM (such as Scalable SQL) which uses both BROUTER.NLM and Btrieve to make remote calls (and which therefore includes the server and volume name when performing an Open operation) has attempted to open a remote file. Because BROUTER.NLM does not interpret the server name, Btrieve attempts to do so but cannot.

92: The transaction table is full.

The maximum number of active transactions was exceeded. Using the Setup utility, specify a higher value for the Number of Transactions configuration option. For more information, refer to "Number of Transactions" on page 65 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

93: The record lock types are incompatible.

The application tried to mix single-record locks (+100/+200) and multiple-record locks (+300/+400) in the same file at the same time. All locks of one type must be released before a lock of the other type can be executed.

94: The application encountered a permission error.

You can receive this status code in the following situations:

- The application tried to open or create a file in a directory without the proper privileges. Btrieve does not override the network privileges assigned to users.

- The designated server is in the server routing table, but your particular workstation is not logged into that server.

- Served-based BREQUEST and client-based Btrieve are trying to access the same file at the same time.

- Btrieve was unable to log in to the NetWare Runtime™ server using the given username. Specifically, one of the following situations exists regarding the supplied username:

  - The user is not a valid user on the Runtime server.

  - The user does not have the appropriate rights to access the file.

  - The username is SUPERVISOR. For security reasons, Btrieve does not allow you to use SUPERVISOR as a username when enabling NetWare runtime support.

95: The network Btrieve session is no longer valid.

The previously established session is no longer active due to an error at the workstation, at the file server, or on the network. Verify that the workstation is still attached to the server, and then unload and reload the Btrieve Requester as discussed in “DOS Requester” in Chapter 4.

This status code can also indicate that the maximum number of sessions for Btrieve has been reached. Use the Communication Statistics option of the Btrieve Monitor utility to see if the maximum number of sessions has been reached. If so, use the Setup utility to specify a higher value for the Number of Remote Sessions configuration option. For more information, refer to "Number of Remote Sessions" on page 67 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

96: A communications environment error occurred.

You tried to attach to Btrieve on a server but the SPX connection table or the Btrieve client table is full. Use the Setup utility to specify a higher value for the Number of Remote Sessions configuration option. For more information, refer to "Number of Remote Sessions" on page 67 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

An NLM application that calls the Btrieve NLM can return Status Code 96 in either of the following situations:

- Not all of the clients have been properly reset.
- The Btrieve NLM was loaded with too small a value for its /s (sessions) parameter. This /s parameter is *not* the same as BSPXCOM’s /s parameter.

97: The network communications buffer is too small.

The application either tried to read or write a record that is longer than 0xFC00 bytes (a Btrieve 6.1*x* internal allowable value), or a record that is longer than the current settings for Btrieve or the Btrieve Requester allow, as follows:

- For an Update, Insert, or Create operation, the application receives this status code if the data buffer length it specifies for the record exceeds Btrieve’s internal communications buffer length.
- For a Get, Step, or Stat operation, the application receives this status code if Btrieve’s internal communications buffer is shorter than the length of the data Btrieve would return, regardless of the data buffer length specified in the application.
- For a Get Chunk or Update Chunk operation, the total size of the retrieved or updated chunk exceeds Btrieve’s internal communications buffer length.

To avoid receiving Status Code 97 in NetWare 4.11*x*, observe the 0xFC00 limit on data size.

In the NetWare Btrieve environment, the effective communications buffer size is the smaller of the values specified for the Largest Record Size option (from the Btrieve Setup utility, and used by BSPXCOM) and the Data Message Length (/D) option (specified when the Btrieve Requester was loaded).

Therefore, to avoid receiving Status Code 97 in this environment, perform one or both of the following steps:

- Increase the size of the communications buffer by using the Btrieve Setup utility to specify a higher value for the Largest Record Size option.
- Reload the Btrieve Requester, and specify a higher value for the Data Message Length (/D) option. For more information, refer to "Data

Message Length (/D)" on page 106 in Chapter 4, “Configuring and Using the Requesters.”

98: Btrieve detected an internal transaction error.

A pre-v6.*x* version of NetWare Btrieve detected an error while executing the operation on a NetWare Transaction Tracking System (TTS) file. The application can perform only an Abort Transaction operation at this point.

Neither v6.*x* NetWare Btrieve nor client-based Btrieve will return Status Code 98, because TTS has a different meaning to v6.*x* NetWare Btrieve, and client-based Btrieve ignores the TTS flag.

99: The Requester cannot access the NetWare Runtime server.

The DOS Requester returns this status code when NetWare Runtime server support is enabled (/C:1) and the Requester either detects no existing connection or cannot find a valid login username. SUPERVISOR is *not* a valid username, even if supplied with the correct password.

If the Requester cannot find a login username other than SUPERVISOR, there is no valid name to pass.

100: No cache buffers are available.

This status code, which can be returned by Btrieve version 6.0 and later, indicates that Btrieve has used all the cache buffers it allocated at load time. Use the Btrieve Setup utility to increase the value for the Cache Allocation configuration option.

If running NetWare Btrieve (v6.0 or later), you also can change the Number of Remote Sessions configuration option to decrease the number of concurrent Btrieve users. For more information, refer to Chapter 3, “Installing and Configuring NetWare Btrieve.”

101: Insufficient operating system memory is available.

This status code, returned by NetWare Btrieve v6.0 and later, indicates that there is not enough operating system memory available to perform the requested operation. To fix this problem, perform one or more of the following:

- Use the Btrieve Setup utility to decrease the value for the Cache Allocation configuration option.

- Use the Btrieve Setup utility to decrease the value of the Number of Remote Sessions configuration option (thereby decreasing the number of concurrent Btrieve users).

- Add memory to the NetWare file server.

For more information on the configuration options, see Chapter 3, “Installing and Configuring NetWare Btrieve.” Also see Status Code 1002, a similar status code that NetWare Communication Services returns in the same situation.

102: Insufficient stack space is available.

This status code, returned as a run-time error by Btrieve v6.0 and later, indicates that Btrieve has run out of stack space. To increase the amount of stack space available to your application, relink the application, setting the stack size to a higher value.

Btrieve returns this message only to NetWare Communication Services applications that call WBTRCALL.DLL, or NLM applications that call Btrieve on the local server.

NetWare Btrieve v6.1*x* requires 3 KB of stack space. If your NLM application receives this status code, try increasing the size of the stack in your application.

You may need more than 3 KB of stack space if you are accessing files defined by Scalable SQL as being under referential integrity (RI) constraints. Because NetWare Btrieve uses recursion when enforcing these constraints, at least 3KB of stack space should be available when you make the Btrieve call.

103: The chunk offset is too big.

This status code, returned by Btrieve 6.0 and later, indicates one of the following possible situations:

- A Get Direct/Chunk operation has specified an offset beyond the end of the record, either explicitly or through the use of the next-in-record bias to the subfunction value. Unless Btrieve returns this status while processing the first chunk, the operation was partially successful. Check the data buffer length parameter immediately after the call to see how much data (and therefore how many chunks) Btrieve retrieved.

- An Update Chunk operation has specified an offset that is more than one byte beyond the end of the record. In this situation, Status Code 103 indicates that Btrieve made no changes to the record.

104: Btrieve does not recognize the locale.

In Btrieve versions 6.0 and later, the Create or Create Index operation returns this status code to indicate that the operating system was not able to return a collation table for the country ID and code page specified. Check that the application specified the locale’s country ID and code page correctly and that the operating system is configured to support the country ID and code page.

105: The file cannot be created with Variable-tail Allocation Tables (VATs).

In Btrieve version 6.0 or later, the application specified that a Btrieve file should be created with Variable-tail Allocation Tables (VATs); however, the application failed to specify that the file was to use variable-length records (a precondition for files to use VATs). This status applies to key-only files as well as regular data files.

106: Btrieve cannot get the next chunk.

In Btrieve version 6.0 or later, the application called the Get Direct/Chunk operation to retrieve a chunk from a record and used the next-in-record bias on the descriptor subfunction. However, after the application established its positioning in the record (but prior to this call), the target record was deleted.

107: The application attempted to perform a chunk operation on a pre-v6.0 file.

In Btrieve version 6.0 or later, the application tried to use either a Get Direct/ Chunk operation or an Update Chunk operation on a pre-6.0 formatted file.

109: The application’s transaction has become too complex.

This status code is returned by NetWare Btrieve (v5.15 and v6.*x*) when a transaction would cause more than 65,535 pages to be written to a pre-image file, and the version of Btrieve that returns the error cannot handle such a large pre-image file.

Break the transaction into two or more smaller transactions.

130: Btrieve ran out of system locks.

This status code is returned by 6.*x* versions of NetWare Btrieve, and can indicate a temporary condition in which no system locks are currently available. For example:

- A single client is performing a very large transaction: one in which thousands of records are being modified.

- Many clients are performing large transactions concurrently.

A client can receive Status Code 130 whether or not it is in a transaction. In some cases, a client can simply retry the failed operation. If other clients have released system locks in the interim, the retried operation may succeed.

If a client in a transaction receives Status Code 130, end or abort the transaction. If the transaction is very large, consider breaking it into multiple smaller transactions.

You can also use the Btrieve Maintenance utility to lower the number of system locks devoted to explicit locking. To do so, lower the values assigned to the Number of Locks and/or Number of Remote Sessions options. See Chapter 3, "Installing and Configuring NetWare Btrieve."

132: The file is full.

NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns this status code.

Client-Based Btrieve for OS/2 and MS Windows Status Codes

Client-based Btrieve may return the following status codes in an OS/2 or MS Windows environment.

1001: The Multiple Locks option is out of range.

In pre-6.*x* client-based versions of Btrieve, the value specified for the Multiple Locks configuration option is out of range. The value must be between 1 and 255, inclusive.

In NetWare 4.11, the value specified for the Number of Locks configuration option is out of range. The value must be between 1 and 64,000, inclusive.

1002: Btrieve cannot allocate the memory needed.

Make sure that the workstation has enough memory to load all the programs it requires.

1003: The Cache Allocation option is too small.

Make sure the value for the Cache Allocation configuration option is large enough to accommodate the required cache size.

1004: The Page Size option is out of range.

The value of the Page Size configuration option must be an even multiple of 512, and it must be between 512 and 4,096, inclusive.

NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

1005: The Pre-Image File Drive option is invalid.

You must specify a valid drive letter for the Pre-Image File Drive configuration option.

Note Pre-image files are used only for files created by Btrieve versions earlier than 6.*x*, *or* by 6.*x* if it was loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option set to Yes.

1006: The Pre-Image Buffer Size option is out of range.

The Pre-Image Buffer Size configuration option must be between 1 and 64, inclusive.

Note Pre-image files are used only for files created by Btrieve versions earlier than 6*x*, *or* by 6.*x* if it was loaded with the Create Btrieve Files in Pre 6.*x* Format configuration option set to Yes.

1007: The Open Files option is out of range.

The Open Files configuration option must be between 1 and 255, inclusive.

1008: The Configuration options are invalid.

The configuration options specified contain invalid or unidentifiable values. For more information on configuration options, see Chapter 3, “Installing and Configuring NetWare Btrieve.”

1009: The Transaction Filename option is invalid.

The filename specified for the Transaction Filename configuration option is not valid. Check to make sure that the transaction filename is correct.

1010: Could not access the transaction control file (BTRIEVE.TRN).

The following situations can cause Btrieve to return this error:

- Btrieve is unable to create BTRIEVE.TRN.
- Btrieve cannot open BTRIEVE.TRN.
- Btrieve cannot read BTRIEVE.TRN.
- Btrieve cannot write to BTRIEVE.TRN (perhaps because BTRIEVE.TRN is flagged read-only).

1011: The Compression Buffer Size specified is out of range.

The Compression Buffer Size configuration option must be between 1 and 64, inclusive.

1013: The task table is full (MS Windows only).

The Btrieve DLL may return this status code if the task entry table is full. You can remedy this situation by increasing the number of available task entries; use the tasks initialization option (tasks=xxx) under the [BTRIEVE] or [BREQUESTDPMI] headings in NOVDB.INI. The minimum value for this option is 1; the maximum value is 64,000.

1014: The application encountered a stop warning.

Btrieve for OS/2 returns this status code when an application calls the **btrvstop** function while files are still open or while a transaction is still active. The application must close all files and end all transactions before calling **btrvstop**.

1015: A pointer parameter is invalid.

One of the pointer parameters passed into Btrieve is invalid. Btrieve will check for invalid pointers (and therefore only return this message) if you put the following line under the [BTRIEVE] heading in your NOVDB.INI file: CHKPARMS=YES. By default, Btrieve performs no pointer checking.

1016: Btrieve is already initialized.

The Btrieve DLL may return this status code if an attempt is made to initialize Btrieve when it is already initialized. To reinitialize Btrieve, close all files, end/ abort all transactions, and call WBTRVSTOP ( ) before calling the initialization function.

1017: The Btrieve Requester for MS Windows cannot find WBTRVRES.DLL.

WBTRCALL.DLL returns this status code when it cannot find the resource file WBTRVRES.DLL. You can remedy this situation by placing a copy of the WBTRVRES.DLL file in the same directory as the WBTRCALL.DLL file.

1018: The application attempted to call Btrieve (either BTRCALL or BTRCALLID) from a Btrieve callback function.

NetWare Communication Services does not allow a Btrieve task to call Btrieve (either BTRCALL or BTRCALLID) from a Btrieve callback function.

1019: The Btrieve engine cancelled the current Btrieve operation at the request of the application's Btrieve callback function.

The application's Btrieve callback function returned a nonzero value, indicating that the application wants to terminate the current Btrieve operation immediately. When the Btrieve engine receives such a cancellation request, it attempts to terminate the currently executing Btrieve operation and ceases to call the callback function for the duration of that operation. Btrieve may be unable to cancel the operation. However, if successful in doing so, Btrieve returns this status code.

Btrieve Requester Status Codes

This section lists the status codes that the Btrieve Requesters may generate.

2001: The memory allocation is insufficient.

In an OS/2 environment, the Requester cannot allocate enough memory for the parameters specified with the BRQPARMS environment variable. In a DOS environment, reduce the value specified for the /D configuration option.

2002: The option is invalid or out of range.

In an OS/2 environment, either one of the options specified with the BRQPARMS environment variable is invalid (such as /P instead of /D) or the value specified for a parameter is out of range. Check the SET BRQPARMS statement to make sure it is correct.

2003: The Requester does not allow local access to the specified file.

The application attempted to access a file stored on a local drive. The version of BTRCALL.DLL installed at the workstation does not allow access to local files.

2004: SPX is not installed.

Install the NetWare SPX 1.3 or later communications software for OS/2.

2005: An incorrect version of SPX is installed.

Install the NetWare SPX 1.3 or later communications software for OS/2.

2006: There is no available SPX connection.

SPX has already established the maximum number of sessions it can handle. To increase the maximum, edit the NET.CFG file. Refer to your NetWare documentation for more information on NET.CFG.

2007: A pointer parameter is invalid.

One of the pointer parameters passed to Btrieve is invalid. Check the program to ensure that the pointer parameters are correct.

Messages

This section lists and describes the messages that Btrieve can return.

The following conventions are used in the descriptions of the messages: *nn* or *xx* refers to a software-supplied number (for example, a status code, operation code, or number of records) and *xxxx* refers to a software-supplied name (for example, a filename, a key type, or a command). Consult the screen message to get the value for *nn*, *xx*, or *xxxx*.

The messages are first arranged alphabetically by their tags (BDIRECT, BREQUEST, BROUTER, BSPXCOM, BTRIEVE, and so on). Within each of these tagged groups, the messages are then arranged numerically by the message numbers within that tagged group.

Novell Directory Services Support Utility Messages

The Novell Directory Services™ (NDS) Support utility sends the following messages, which apply only to Btrieve 6.1*x* for NetWare 4.

BDIRECT-6.10-6: The Directory Services current context could not be established.

The NDS Support utility returns this message if the bindery context has not been set. For information on how to set the bindery context, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6.10-7: The Directory Services error *nn* was returned while checking the Btrieve object status.

For information about the error code returned by Directory Services, refer to the NetWare *System Messages* manual.

BDIRECT-6.10-15: The Directory Services error *nn* occurred while attempting to log in; please try again.

For information about the error code returned by Directory Services, refer to the NetWare *System Messages* manual.

BDIRECT-6.10-20: The Directory Services error *nn* was returned while installing the Btrieve object.

For information about the error code returned by Directory Services, refer to the NetWare *System Messages* manual.

BDIRECT-6.10-22: The Directory Services error *nn* was returned while adding the Btrieve object to the server resource attribute list. This will not affect the Btrieve Requester, but other communication software that uses this attribute may not function correctly.

For information about the error code returned by Directory Services, refer to the NetWare *System Messages* manual.

BDIRECT-6.10-27: The Directory Services error *nn* was returned while removing the Btrieve object.

For information about the error code returned by Directory Services, refer to the NetWare *System Messages* manual.

BDIRECT-6.10-32: The Btrieve object could not be activated.

The NDS Support utility returns this message if the Btrieve object has not been created or cannot be found. Ensure that you have specified the correct object name. For information on using the NDS Support utility to install a Btrieve object, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6.10-33. The Btrieve object was activated successfully.

This is an informational message returned by the NDS Support utility. No action is required.

BDIRECT-6.10-34: The Btrieve object could not be deactivated.

The NDS Support utility returns this message if the Btrieve object has not been activated or cannot be found. Ensure that you have specified the correct object name. For information on using the NDS Support utility to install or remove a Btrieve object, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6.10-35: The Btrieve object was deactivated successfully.

This is an informational message returned by the NDS Support utility. No action is required.

BDIRECT-6.10-37: A file error occurred accessing the BSTART.NCF file. The Btrieve object may not be activated or deactivated.

For information on using the NDS Support utility to install or remove a Btrieve object, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6-10-38: A file error occurred accessing the BSTOP.NCF file. The Btrieve object may not be activated or deactivated.

For information on using the NDS Support utility to install or remove a Btrieve object, see the section titled, "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6-10-39: A file access error occurred while checking the Btrieve BSTART and BSTOP NCF files. These files must be copied to the SYS:SYSTEM directory before the Btrieve object may be installed or removed.

The NDS Support utility returns this message if you do not have access rights to the SYS:SYSTEM directory. For information on using the NDS Support utility to install or remove a Btrieve object, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

BDIRECT-6-10-40: A file error occurred accessing the BDIRECT.TMP file. This file is used to modify the BSTART and BSTOP NCF files. The Btrieve object may not be activated or deactivated correctly when using the NCF files.

The NDS Support utility returns this message if you do not have access rights to the SYS:SYSTEM directory. For information on using the NDS Support utility to install or remove a Btrieve object, see the section titled "Registering NetWare Btrieve with the Novell Directory Services" on page 93 in Chapter 3, “Installing and Configuring NetWare Btrieve.”

Btrieve Rebuild Utility Messages

The following messages are sent by the Rebuild utility (BREBUILD.NLM).

BREBUILD–1.1–2: BREBUILD could not allocate memory.

The Rebuild utility displays this message when the server does not have adequate memory. Acquire more memory for the server.

BREBUILD–1.1–3: BREBUILD could not rename *xxxx* to *xxxx*.

The Rebuild utility displays this message when the specified file does not exist. Check to see if the file exists.

BREBUILD–1.1–4: An incorrect Btrieve version is loaded.

The Rebuild utility displays this message when an incorrect Btrieve version is loaded. Unload the current version of Btrieve, and reload using Btrieve 6.*x*.

BREBUILD–1.1–5: BREBUILD could not delete *xxxx*.

The Rebuild utility displays this message when it cannot delete the specified file. Check to see if the file is open. If it is open, close it. The Rebuild utility cannot delete a file while it is open.

BREBUILD–1.1–6: An internal program error occurred.

The Rebuild utility detected an internal diagnostic error.

BREBUILD–1.1–7: BREBUILD could not open *xxxx*. The Btrieve status was *nn*.

The Rebuild utility displays this message when it cannot open the specified file. Btrieve returns the specified status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BREBUILD–1.1–8: BREBUILD could not open *xxxx* in the accelerated mode. The Btrieve status code was *nn*.

The Rebuild utility displays this message when it cannot open the specified file in Accelerated mode. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BREBUILD–1.1–9: BREBUILD could not create the new file or files with the Btrieve 6.*x* advanced features. Ensure that the correct version of Btrieve is loaded with the Create Btrieve Files in Pre 6.*x* Format option set to No.

The Rebuild utility displays this message when it cannot create the new Btrieve files in 6.*x* format. Unload Btrieve by executing `BSTOP`, or unload Btrieve at the server console or at a workstation running RCONSOLE.

BREBUILD–1.1–10: An invalid option was specified.

Change the configuration options for the Rebuild utility through the Setup utility. Refer to "Configuring NetWare Btrieve" on page 63 in Chapter 3, “Installing and Configuring NetWare Btrieve” for more information. When you run the Rebuild utility interactively, it checks the values you enter to ensure they are within the proper ranges.

BREBUILD–1.1–11: Indexes are not consecutively numbered. Indexes will not be dropped.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–12: BREBUILD has copied *nn* records so far.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–13: BREBUILD is rebuilding *nn* key number *nn*.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–14: BREBUILD did not rebuild for the following reason: *xxxx*

The Rebuild utility was not able to rebuild the file specified and returned this message. Eliminate the cause for this message and try to rebuild the file again.

BREBUILD–1.1–15: Btrieve returned status *nn* for the following reason: *xxxx*

Btrieve returned the specified status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BREBUILD–1.1–16: The *xxxx* file is already in 6.*x* format.

The Rebuild utility displays this message when the specified file is already in Btrieve 6.*x* format. You do not need to rebuild the file.

BREBUILD–1.1–17: The *xxxx* file is not a Btrieve file.

The file you specified to rebuild is not a valid Btrieve file. The Rebuild utility cannot rebuild the file.

BREBUILD–1.1–18: BREBUILD could not obtain the characteristics of the *xxxx*. The Btrieve status was *nn*.

The Rebuild utility returns the specified Btrieve status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BREBUILD–1.1–19: BREBUILD could not change to directory *xxxx*.

The directory name you specified for the output directory is not valid. Specify a valid output directory name for the Rebuild utility on the command line or through the Setup utility.

BREBUILD–1.1–20: BREBUILD is processing *xxxx*.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–21: BREBUILD is copying the *xxxx* to the temporary file *xxxx*.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–22: The file was rebuilt successfully.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–23: The utility could not open sys:\system\brebuild.log.

The Rebuild utility displays this message if the file BREBUILD.LOG either does not exist or has been left open.

BREBUILD–1.1–24: An error occurred accessing *xxxx*.

The Rebuild utility displays this message when you specify an invalid filename. Make sure you specified a valid filename.

BREBUILD–1.1–25: BREBUILD is setting the owner name of an empty target file.

The Rebuild utility displays this message when an error occurs during the setting of an empty target file's owner name. You need to rerun the Rebuild utility.

BREBUILD–1.1–26: BREBUILD is dropping the indexes of an empty target file.

The Rebuild utility displays this message when an error occurs during the dropping of an empty target file's indexes. You need to rerun the Rebuild utility.

BREBUILD–1.1–27: BREBUILD is reading the first record from the old file.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–28: BREBUILD is inserting records into the new file.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–29: BREBUILD is reading records from the old file.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–30: BREBUILD is putting back indexes on the new file.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–31: BREBUILD could not create *xxxx*. The Btrieve status code was *nn*.

The Rebuild utility displays this message when it cannot create the specified file. The utility returns the specified Btrieve status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section "Status Codes" in this appendix.

BREBUILD–1.1–32: BREBUILD copied a total of *nn* records.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–33: BREBUILD start time is *nn*.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–34: Key number *nn* is invalid.

The Rebuild utility displays this message when the key specified for the Key Number (–K) option is invalid. Specify a valid key number for the Rebuild utility on the command line or through the Setup utility.

BREBUILD–1.1–35: Page size *nn* is invalid.

The Rebuild utility displays this message when the page size specified for the Page Size (–P) option is invalid. Specify a valid page size for the Rebuild utility on the command line or through the Setup utility.

NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

BREBUILD–1.1–36: The page size will be changed to *nn.*

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–37: The rebuild process has ended.

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–38: BREBUILD was terminated by the user.

The user entered Ctrl+C and the Rebuild utility was unloaded.

BREBUILD–1.1–39: BREBUILD was unloaded by the user.

The user unloaded Btrieve and the Rebuild utility was unloaded.

BREBUILD–1.1–40: BREBUILD could not open the file *xxxx*.

The Rebuild utility displays this message when it cannot open the specified file. The file may be open or may not exist. Check for one of these conditions.

BREBUILD–1.1–42: The specified local server *xxxx* is invalid.

The Rebuild utility displays this message when the specified server name is not a local server.

BREBUILD–1.1–44: BREBUILD could not clone *xxxx*. The Btrieve status was *nn*.

The Rebuild utility displays this message when it cannot clone the specified file. Btrieve returns the specified Btrieve status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section "Status Codes" in this appendix.

BREBUILD–1.1–45: The command line file *xxxx* does not have an <end> delimiter.

Each entry in the command file contains the utility options (if any) and the set of files to convert. Each entry must be followed by <end> or [end]. Edit the command file to include the necessary delimiter.

BREBUILD–1.1–48: BREBUILD could not initialize the user interface library.

The Rebuild utility displays this message when the server has insufficient memory to initialize the user interface library.

BREBUILD–1.1–63: BREBUILD could not initialize localized message table.

The Rebuild utility displays this message when the server has insufficient memory to initialize the localized message table.

BREBUILD–1.1–64: BREBUILD is currently processing the following file:

This is an informational message from the Rebuild utility. No action is required.

BREBUILD–1.1–66: BREBUILD could not open the *xxxx*.

The Rebuild utility displays this message when it cannot open the specified file. The file may be open or may not exist. Check for one of these conditions.

Btrieve Requester Messages

The following messages are sent by the Btrieve DOS Requester (BREQUEST.EXE).

BREQUEST–6.1–1: The message file *xxxx* is invalid; BREQUEST cannot be loaded.

The Btrieve Requester returns this message when the message file is invalid. Specify a valid message file so that BREQUEST.EXE can be loaded.

BREQUEST–6.1–3: The option specified is not a valid option.

The Btrieve Requester returns this message when the option specified is not a valid option. Specify a valid option.

BREQUEST–6.1–4: The value specified for the Data Message Length (/d) option is invalid.

The Btrieve Requester returns this message when the value specified for the Data Message Length (/D) option is invalid. Specify the /D option as /D:n, where *n* is an integer.

BREQUEST–6.1–5: The workstation has insufficient memory to load BREQUEST.

The Btrieve Requester returns this message when the workstation has insufficient memory to load the Requester. Unload unnecessary programs or try a smaller value for the /D parameter.

BREQUEST–6.1–6: BREQUEST is already loaded.

The Btrieve Requester returns this informational message to indicate that BREQUEST.EXE is already loaded.

BREQUEST–6.1–7: XQL or NSREQ is already loaded; BREQUEST must be loaded first.

The Btrieve Requester returns this message when XQL or NSREQ is already loaded; the DOS Requester must be loaded first. Unload XQL or NSREQ and then load the DOS Requester.

BREQUEST–6.1–8: DOS 2.00 or greater is not loaded; load DOS 2.00 or greater.

The Btrieve Requester returns this message. DOS 2.*x* or later is not loaded; you must load DOS 2.*x* or later to proceed.

BREQUEST–6.1–9: The SPX.COM file is not loaded; load the file SPX.COM.

The Btrieve Requester returns this message when the file SPX.COM is not loaded. Load SPX.COM.

BREQUEST–6.1–10: The function SPXInitialize returned an error. Make sure the file IPX.COM is loaded.

The Btrieve Requester returns this message when the SPXInitialize function encounters an error. You must make sure that the file IPX.COM is loaded**.**

BREQUEST–6.1–11: The IPX socket table is full.

The Btrieve Requester returns this message when the IPX socket table is full.

BREQUEST–6.1–13: The value specified for the Runtime server support (/C) option is invalid.

The Btrieve Requester returns this message when an invalid value is specified for the NetWare Runtime Server Support option (/C). Specify this option in one of these forms:

/C:0 To disable NetWare Runtime server support.

/C:1 To enable NetWare Runtime server support.

To authenticate requests on the NetWare Runtime server, provide a username and password, separating them with commas, as follows:

/C:1,*`username`*,*`password`*

For more information about this option, see Chapter 4, "Configuring and Using the Requesters."

Btrieve Requester Utility Messages

The following messages are sent by the Requester utility (BREQUTIL.NLM).

BREQUTIL–6.1–3: The single-user version of Btrieve v*n.n* is loaded.

This is an informational message from the Requester utility. No action is required.

BREQUTIL–6.1–4: The network version of Btrieve v*n.n* is loaded.

This is an informational message from the Requester utility. No action is required.

BREQUTIL–6.1–5: The multi-user version of Btrieve v*n.n* is loaded.

This is an informational message from the Requester utility. No action is required.

BREQUTIL–6.1–6: The OS/2 DynaLink for Btrieve v*n.n* is loaded.

This is an informational message from the Requester utility. No action is required.

BREQUTIL–6.1–7: Btrieve version v*n.n* is loaded.

This is an informational message from the Requester utility. No action is required.

BREQUTIL–6.1–8: Btrieve is not loaded.

Either the Btrieve Requester (BREQUEST.EXE) or Record Manager (BTRIEVE.EXE) is not loaded.

BREQUTIL–6.1–9: Btrieve operation *xx* was unsuccessful. The number of the applicable Btrieve status code is *nn*.

The Requester utility returns the specified Btrieve status code. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BREQUTIL–6.1–16: Btrieve cannot be removed from memory while Scalable SQL is loaded.

This is an informational message from the Requester utility. No action is required.

Btrieve Roll Forward Utility for DOS

This section lists and describes the messages that the Roll Forward utility for the DOS environment can send.

BROLLFWD–6.1–1: This workstation has insufficient memory to complete the operation.

The workstation does not have enough memory for the DOS Roll Forward utility to complete the current operation. Free some memory on the workstation and restart the Roll Forward process.

BROLLFWD–6.1–2: The log file was created at *nn:nn*.

This is an informational message from the DOS Roll Forward utility. No action is required.

BROLLFWD–6.1–3: An internal program error has occurred.

The DOS Roll Forward utility detected an internal program error.

BROLLFWD–6.1–4: The specified owner name is invalid. Please reenter the owner name:

The DOS Roll Forward utility returns this message when the specified owner name is invalid. Specify a valid owner name and start the roll forward process again. The maximum length for an owner name is 8 bytes. Enter a valid owner name.

BROLLFWD–6.1–21: The utility is unable to open *xxxx*. The number of the applicable system error code is *nn*.

The DOS Roll Forward utility returns this message when it cannot open the specified file. The file may be open or may not exist.

BROLLFWD–6.1–22: An error occurred while the utility was reading *xxxx*.

The DOS Roll Forward utility returns this message when it cannot read the specified file. Verify that the file exists. If the file exists, ensure that the file is closed.

BROLLFWD–6.1–23: File *xxxx* ended unexpectedly during the roll forward process.

The DOS Roll Forward utility returns this message when the specified file ended unexpectedly during the roll forward process. The associated log file is corrupt. You cannot use this log file.

BROLLFWD–6.1–24: The utility could not find the log file header information in *xxxx*.

The DOS Roll Forward utility returns this message when no log file header information was found for the specified file. The associated log file is corrupt. You cannot use this log file.

BROLLFWD–6.1–25: The *xxxx* file contains incorrect syntax in the log file header.

The DOS Roll Forward utility returns this message when the log file header for the specified file contains incorrect syntax. The associated log file is corrupt. You cannot use this log file.

BROLLFWD–6.1–26: Btrieve operation *nn* was unsuccessful. The number of the applicable Btrieve status code is *nn*.

Btrieve returns the specified Btrieve operation and status codes to the DOS Roll Forward utility. The specified Btrieve operation was unsuccessful. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix. For more information about the Btrieve operation *nn*, refer to the *Btrieve Programmer’s Manual*.

BROLLFWD–6.1–27: The utility could not find *xxxx* in the BLOG.CFG file.

The DOS Roll Forward utility returns this message when the specified file is not found in the log configuration file, BLOG.CFG. Edit the BLOG.CFG file to add the filename specified. For more information about the BLOG.CFG file, refer to the section “Creating the Log Configuration File” in Chapter 5.

BROLLFWD–6.1–28: The record size of the Btrieve file *file xxxx* exceeds the size of the data buffer specified with the /d parameter.

The DOS Roll Forward utility returns this message when the value specified for the /D option is invalid. Increase the value for the /D option.

BROLLFWD–6.1–29: The length of the data to be printed, specified with the /t option, is invalid. Valid data lengths range from 1 through the value of the data buffer size specified with the /d option.

The DOS Roll Forward utility returns this message when the value specified for the /T option is invalid. Change the value for this option.

BROLLFWD–6.1–30: The length of the key to be printed, specified with the /k option, is invalid. Valid lengths for printing keys range from 1 through 255 characters.

The DOS Roll Forward utility returns this message when the value specified for the /K option is invalid. Enter a value in the range 1 through 255.

BROLLFWD–6.1–31: 0/1 entry in the log file has been processed.

This is an informational message from the DOS Roll Forward utility. No action is required.

BROLLFWD–6.1–32: *xx* in the log file have been processed.

This is an informational message from the DOS Roll Forward utility. No action is required.

BROLLFWD–6.1–42: The disk is full.

The disk is full. Free some space on the disk and run the DOS Roll Forward utility again.

BROLLFWD–6.1–43: The utility started the roll forward process at *nn:nn*.

This is an informational message from the DOS Roll Forward utility. No action is required.

Btrieve Message Router Messages

The following messages are sent by the Btrieve Message Router (BROUTER.NLM or BDROUTER.NLM).

Note In this appendix, all references to BROUTER also apply to BDROUTER.

BROUTER–6.1–1: The server has insufficient memory to execute BROUTER.

BROUTER returns this message when the server has insufficient memory to load the file BROUTER.NLM. Free some memory by unloading NLMs or reconfiguring NLMs to use less memory.

BROUTER–6.1–2: The value specified for a configuration option is invalid.

BROUTER returns this message when the value specified for a configuration option is invalid. Reload BROUTER.NLM using valid options.

BROUTER–6.1–3: An internal error has occurred; the SPXOpenSocket function failed.

BROUTER returns this message if an internal diagnostic error occurs. The SPXOpenSocket function failed. Another NLM may be using the socket number reserved for BROUTER. If you receive this message, unload all other NLMs and then load BTRIEVE.NLM and BROUTER.NLM. Finally, reload the other NLMs. This process reveals the NLM that is using BROUTER's socket number.

BROUTER–6.1–6: BROUTER is loaded.

BROUTER returns this message to indicate that BROUTER.NLM is now loaded.

NetWare Btrieve Setup Utility Messages

The following messages are sent by the Setup utility (BSETUP.NLM).

BSETUP–6.1–40: The Btrieve Setup utility was unable to save the file.

The Setup utility was not able to save the Btrieve configuration options to the BSTART.NCF file. Retry the operation.

BSETUP–6.1–43: The attempted memory allocation failed.

The Setup utility cannot allocate enough internal memory to display the Rebuild log file.

BSETUP–6.1–63: The BSTART.NCF file is incomplete or invalid.

The Setup utility is unable to save the configuration option changes to BSTART.NCF because the file is incomplete or invalid. You may want to delete BSTART.NCF and recreate it in the Setup utility.

BSETUP–6.1–73: The Btrieve Setup utility could not find the BSTART.NCF file, which must reside in the SYS:SYSTEM directory on the server.

The Setup utility cannot find the BSTART.NCF file. If BSTART.NCF does not exist, you must create it through the Setup utility.

BSETUP–6.1–75: To implement the configuration changes, first unload BSPXCOM.NLM and BTRIEVE.NLM, and then use the BSTART.NCF file to reload BSPXCOM.NLM and BTRIEVE.NLM.

For the changes you made to the configuration options to take effect, you must unload BSPXCOM.NLM and BTRIEVE.NLM and then reload them using BSTART.NCF.

BSETUP–6.1–100: The BSTART.NCF file was created with the default configuration settings. To view or change these settings, select the Set Btrieve Configuration option on the Available Options menu.

Select the stated option within the Setup utility to change the configuration options for Btrieve in the BSTART.NCF file.

BSETUP–6.1–121: WARNING: Before running the Btrieve Rebuild utility, be sure to back up all your Btrieve data files.

Before running the Rebuild utility, make sure you have backed up all your Btrieve data files. Having a backup copy ensures against data loss if a power interruption occurs while you are running the utility. Another reason it is better to keep a backup is that the Rebuild utility deletes the original Btrieve files.

BSETUP–6.1–122: You must specify the Rebuild Configuration parameters before you can execute the Rebuild utility.

Before you can run the Rebuild utility, you must specify appropriate values for the Rebuild configuration options in the Setup utility. Please do so before attempting to run the Rebuild utility again.

BSETUP–6.1–131: The Output Directory pathname does not require a file specification.

Do not specify a filename as part of the output directory path.

BSETUP–6.1–132: Before rebuilding Btrieve files on a remote server, make sure the following files are loaded: BROUTER.NLM and BTRIEVE.NLM on the local server, and BSPXCOM.NLM and BTRIEVE.NLM on the remote server.

The NLMs™ listed in the message must be loaded on a local server and a remote server before attempting to rebuild files on a remote server.

BSETUP–6.1–148: The Btrieve Setup utility could not find the file BREBUILD.NLM in the system directory, so you cannot invoke the Rebuild utility at this time.

The Setup utility returns this message when it cannot find the file BREBUILD.NLM. The file BREBUILD.NLM must be in the system directory. Please verify that it is in the correct directory.

BSETUP–6.1–152: Either the Input Directory pathname is invalid, or no matching files could be found. Please check the pathname.

The Setup utility returns this message when the path specified for the Files To Be Converted field is invalid or when no files exist in the directory specified.

BSETUP–6.1–153: Either the Output Directory pathname is invalid, or no matching files could be found. Please check the pathname.

The Setup utility returns this message when the path specified for the Output Directory field does not exist on the local server.

BSETUP–6.1–158: The Btrieve Setup utility could not create a screen for the Rebuild utility, so you cannot invoke the Rebuild utility at this time.

The Setup utility cannot invoke the Rebuild utility because too many screens are active at this time. Try the Rebuild utility again later when there is less activity on the server.

BSETUP–6.1–159: The Btrieve Setup utility could not open the Rebuild error log file.

When you attempted to view the Rebuild log file, the Setup utility was not able to open the log file. Verify that the file exists before retrying the operation.

BSETUP–6.1–160: The attempt to read the Rebuild error log file failed.

The Setup utility was not able to read the Rebuild log file when you attempted to view it. Check to see that the log file exists.

BSETUP–6.1–162: The Rebuild utility cannot be loaded because the BTRIEVE.NLM is not loaded. Please load Btrieve using the BSTART.NCF file.

Btrieve must be loaded before you run the Rebuild utility. Load BTRIEVE.NLM using BSTART.NCF.

BSETUP–6.1–166: The utility found no files in the specified directory path. Please check the pathname.

The Setup utility cannot find any Btrieve files to rebuild in the chosen directory (specified for Files To Be Converted). Verify that you specified the correct directory.

BSETUP–6.1–167: The Rebuild utility could not be invoked at this time.

The Setup utility returns this message when it cannot invoke the Rebuild utility. Check to make sure that the version 1.1 of BREBUILD.NLM is loaded at the server. If it is not, load it at this time.

BSETUP–6.1–168: The Btrieve Setup utility could not create the semaphore which synchronizes BSETUP.NLM and BREBUILD.NLM, so you cannot invoke the Rebuild utility at this time.

The semaphore used to synchronize the Setup utility and the Rebuild utility could not be created. Try the utility at a later time when there is less activity at the file server or use the command line version of BREBUILD.

Btrieve SPX Communications Messages

The following messages are sent by the Btrieve SPX communications module (BSPXCOM.NLM).

BSPXCOM–6.1–1: The option specified is not a valid option.

The Btrieve SPX communications module returns this message when the option specified is not a valid option. Specify a valid option.

BSPXCOM–6.1–2: The server has insufficient memory to execute BSPXCOM.

The Btrieve SPX communications module returns this message if the server has insufficient memory to load the file BSPXCOM.NLM. If you receive this message, you must free memory by unloading NLMs or reconfiguring NLMs to use less memory.

BSPXCOM–6.1–3: An internal error has occurred. BSPXCOM detected a semaphore allocation failure.

The Btrieve SPX communications module returns this message when an internal diagnostic error occurs. BSPXCOM detected a semaphore allocation failure.

BSPXCOM–6.1–4: The Service Request Block (SRB) function code *nn* contains invalid data. Check for an incompatible version of the file BSPXCOM.NLM.

The Btrieve SPX communications module returns this message if the Service Request Block (SRB) function code contains invalid data. Check that BSPXCOM’s version is compatible with the version number of the workstation’s Btrieve Requester.

BSPXCOM–6.1–6: An internal error has occurred. The session was not found in ConnTable.

The Btrieve SPX communications module returns this message when an internal diagnostic error occurs. The session was not found in ConnTable.

BSPXCOM–6.1–7: Another NLM is using the socket number reserved for BSPXCOM.

The Btrieve SPX communications module detected another NLM using the socket number reserved for BSPXCOM. Unload all other NLMs and then load BTRIEVE.NLM and BSPXCOM.NLM. Finally, reload the other NLMs. This process reveals the NLM that is using BSPXCOM's socket number.

BSPXCOM–6.1–8: An SPX-level receive I/O error (`hexadecimal code nn`) has occurred. The connection has been lost.

The Btrieve SPX communications module returns this message when an SPX-level receive I/O error occurs. The connection has been lost.

BSPXCOM–6.1–9: An SPX-level send I/O error (`hexadecimal code nn`) has occurred. The connection has been lost.

The Btrieve SPX communications module returns this message when an SPX-level send I/O error occurs. The connection has been lost.

BSPXCOM–6.1–10: An internal error (`hexadecimal code nn`) has occurred. BSPXCOM detected a Dequeue error.

The Btrieve SPX communications module returns this message when an internal diagnostic error (a dequeue error) occurs.

BSPXCOM–6.1–11: Bad connection ID detected on receive. The SPX connection was lost after the initial request began.

The Btrieve SPX communications module detected a bad connection ID on a receive. The SPX connection was lost after the initial request began. No action is needed if this message appears occasionally when you reboot your workstation.

However, if this message appears frequently when you have not rebooted your workstation, you must increase your workstation's SPX Timeout parameter. Also, you can check for NLMs monopolizing the CPU time.

BSPXCOM–6.1–12: Bad connection ID detected on send. The SPX connection was lost after the initial request began.

The Btrieve SPX communications module returns this message when it detects a bad connection ID on a send. The SPX connection was lost after the initial request began. No action is needed if this message appears occasionally when you reboot your workstation.

However, if this message appears frequently when you have not rebooted your workstation, you must increase your workstation's SPX Timeout parameter. Also, you can check for NLMs monopolizing the CPU time or for a hardware failure.

BSPXCOM–6.1–13: An error (`hexadecimal code nn`) was detected while trying to establish an SPX session requested by a remote workstation.

The Btrieve SPX communications module detected an internal diagnostic error while it was trying to establish an SPX connection requested by a remote workstation.

BSPXCOM–6.1–15: The request for statistics from the Btrieve Monitor utility was not recognized. Check for an incompatible version of the utility or BSPXCOM.

The Btrieve SPX communications module returns this message if the request for statistics from the Btrieve Monitor utility was not recognized. Check for an incompatible version of the utility or BSPXCOM.

BSPXCOM–6.1–16: The session was rejected; the session limit was reached. Increase the value specified for the Number of Remote Sessions option.

The Btrieve SPX communications module returns this message to indicate the session was rejected because the session limit was reached. Increase the value specified for the Number of Remote Sessions configuration option. Unload and then reload BSPXCOM.NLM so that the new value can be used. (For more information about this option, refer to "Number of Remote Sessions" on page 67 in Chapter 3, “Installing and Configuring NetWare Btrieve.”)

BSPXCOM–6.1–17: An internal error has occurred. BSPXCOM did not recognize the GET_EIM_STATS function.

The Btrieve SPX communications module returns this message when an internal diagnostic error occurs. BSPXCOM did not recognize the GET_EIM_STATS function. Check to see if the version of BSPXCOM is compatible with that of the Btrieve Monitor utility (BTRMON.NLM).

Btrieve NLM Messages

The messages in this section are sent by the Btrieve NLM.

BTRIEVE–6.1–1: The value specified for the Cache Allocation option is invalid.

The Btrieve NLM returns this message when the value specified for the Cache Allocation option is invalid. Use the Setup utility and specify a value between 32 through 64,000 for this option.

BTRIEVE–6.1–2: The value specified for the Largest Compressed Record Size option is invalid.

The Btrieve NLM returns this message when the value specified for the Largest Compressed Record Size option is invalid. Return to the Setup utility and specify a valid value for the Largest Compressed Record Size option.

BTRIEVE–6.1–4: The value specified for the Number of Open Files option is invalid.

The Btrieve NLM returns this message when the value specified for the Number of Open Files option is invalid. Use the Setup utility and specify a value between 1 and 64,000 for this option.

BTRIEVE–6.1–7: The value specified for the Number of Handles option is invalid.

The Btrieve NLM returns this message when the value specified for the Number of File Handles option is invalid. Use the Setup utility and specify a value between 1 and 64,000 for this option.

BTRIEVE–6.1–9: The value specified for the Number of Remote Sessions option is invalid.

The Btrieve NLM returns this message when the value specified for the Number of Remote Sessions option is invalid. Use the Setup utility and specify a value between 1 and 64,000 for this option.

BTRIEVE–6.1–10: Not enough memory is available for merging and sorting.

The Btrieve NLM returns this message when not enough memory is available for Btrieve to perform its internal sort/merge routines.

To provide Btrieve with more memory to perform these routines, increase the value of `/q:` for the options keyword under the `[Btrieve Client]` section of your NOVDB.INI file, then restart Btrieve. If no `/q:` exists, add it, and specify a value greater than 64KB (`/q=64`).

When using Btrieve callback routines in NetWare 4.11, always specify a value for the `/q:` option in order to reserve some memory for use by your application's callback function. Reserve at least 1MB of memory (`/q=1024`) for the callback's use.

BTRIEVE–6.1–12: The value specified for the *–option* option is invalid.

The Btrieve NLM returns this message when the value for an option is not valid. Return to the BSTART.NCF file and enter the correct value.

BTRIEVE–6.1–13: The option specified is not a valid option.

The Btrieve NLM returns this message when the option specified is not a valid option. Return to the BSTART.NCF file and remove the invalid option.

BTRIEVE–6.1–14: Continuous operation is active on one or more files. Roll-in could not be completed. Please free some disk space if you wish to end continuous operation before unloading the BTRIEVE NLM.

The Btrieve NLM returns this message when the server has insufficient disk space to allow Btrieve to complete the roll in. Free some disk space in order to end continuous operation and unload the Btrieve NLM.

Client-based versions of the Btrieve engine do not return this message because they do not support continuous operation.

BTRIEVE–6.1–16: The server has insufficient memory to complete the operation.

The Btrieve NLM returns this message when the server has insufficient memory to allow Btrieve to load as it is configured. Use the Setup utility to reconfigure Btrieve to use less memory, or unload any unnecessary NLMs.

BTRIEVE–6.1–17: The header in log file *xxxx* is invalid.

The Btrieve NLM returns this message when the header in log file *xxxx* is invalid. The file *xxxx* is expected to be a log file according to BLOG.CFG but it is not. If you receive this message, either delete file *xxxx* and create a new log file, or change BLOG.CFG to specify a different log file.

BTRIEVE–6.1–18: The log file *xxxx* cannot be created in the location specified.

The Btrieve NLM returns this message when log file *xxxx* cannot be created in the location specified. Check that the disk is not full and that the user has rights to create and write to log file *xxxx*.

BTRIEVE–6.1–19: Logging is not active for file *xxxx*; check the file specification in BLOG.CFG.

The Btrieve NLM returns this message when logging is not active for the file *xxxx*; check the file specification in BLOG.CFG. This message is always displayed with the `Invalid Header in Log File` *xxxx* message or the `Unable to Create Log File` *xxxx* message.

BTRIEVE–6.1–20: The log file *xxxx* cannot be written. Check disk space.

The Btrieve NLM returns this message when log file *xxxx* cannot be written. Check the disk space. If the disk is full, free some space by deleting any unnecessary files.

BTRIEVE–6.1–21: Logging has stopped for file *xxxx*.

The Btrieve NLM returns this message when logging has stopped for file *xxxx*. The log file *xxxx* cannot be written. Check the disk space. If the disk is full, free some space by deleting unnecessary files.

BTRIEVE–6.1–22: Files found in continuous operation are being rolled in.

The Btrieve NLM returns this message when files have been found in continuous operation. The Btrieve NLM then rolls in (updates) those files.

BTRIEVE–6.1–23: An error occurred while Btrieve was rolling in files that are in continuous operation.

The NetWare Btrieve engine detected files in continuous operation, and while rolling in (updating) these files, the engine detected an error. Check to see if the disk is full or if any other disk problems exist.

BTRIEVE–6.1–24: Files have been found in continuous operation; these files have been successfully rolled in.

The Btrieve NLM detected files in continuous operation and rolled in (updated) these files successfully.

BTRIEVE–6.1–25: The file *xxxx* is rolling back.

The Btrieve NLM returns this message when the file *xxxx* is rolling back. (Rolling back refers to aborting a transaction and undoing all changes made to the database during the transaction, thus restoring the database to the state it was in before the transaction began.)

BTRIEVE–6.1–26: The transaction roll back cannot be completed. Check the memory available on the server.

The Btrieve NLM returns this message when it is rolling back the transaction but the transaction roll back cannot be completed. (Rolling back refers to aborting a transaction and undoing all changes made to the database during the transaction, thus restoring the database to the state it was in before the transaction began.)

Check the memory available on the server. Free memory by unloading NLMs or reconfiguring NLMs to use less memory.

BTRIEVE–6.1–27: The transaction control file cannot be created.

The Btrieve NLM returns this message when it cannot create or open the transaction control file (BTRIEVE.TRN). Make sure that no user or client has that file open, and make sure that BTRIEVE.TRN resides in the SYS:\SYSTEM directory.

Btrieve Monitor Utility Messages

The following messages are sent by the Btrieve Monitor utility (BTRMON.NLM).

BTRMON–6.1–43: The utility cannot allocate memory sufficient for the operation.

Press Esc to continue. The Btrieve Monitor utility makes five attempts to allocate enough memory for the operation and, if unsuccessful, displays message 118.

BTRMON–6.1–118: There is still insufficient memory for the operation.

The Btrieve Monitor utility displays this message when it has completed five unsuccessful attempts to allocate enough memory for the operation. You must either unload some NLMs, if possible, or wait for a time when server usage is less heavy.

BTRMON–6.1–124: The utility cannot find the help file BTRMON.HLP.

The Btrieve Monitor utility displays this message when it cannot find the help file. Check your SYSTEM directory to make sure that the help file is present.

BTRMON–6.1–128: This Btrieve file is no longer open. Please press the <Insert> key to update the file list.

The Btrieve Monitor utility displays this message when you select a file that was open when you began viewing active resources but is now closed.

Btrieve Maintenance Utility Messages

The following messages are sent by the Btrieve Maintenance utility (BUTIL.NLM).

BUTIL–6.10–1: The keyword compression is no longer supported. Use truncate.

The compress keyword is no longer used in description files. However, you may reduce the file size for files that contain variable-length records by specifying *y* for the truncate keyword. The truncate keyword has no effect unless the file contains variable-length records.

BUTIL–6.10–2: An error occurred while BUTIL was accessing the description file.

The Maintenance utility returns this message when an error occurs during access of a description file. Make sure the file has not been corrupted and that the information it contains is properly formatted.

BUTIL–6.10–6: The BUTIL command is invalid.

The Maintenance utility returns this message when the syntax of the command you entered is incorrect. Verify the syntax before reentering the command.

BUTIL–6.10–8: The command completed, but one or more errors occurred.

An error occurred when you executed a command that performed a number of Btrieve operations. These commands include COPY, LOAD, or CLONE. This message is accompanied by additional messages that can help you identify the problem.

BUTIL–6.10–9: The command did not complete due to an unrecoverable error.

The Maintenance utility returns this message when the command you entered did not complete successfully due to an unrecoverable error. Verify that the syntax you entered is correct before reentering the command. This message is accompanied by additional messages that can help you identify the problem.

BUTIL–6.10–10: The command line contains a syntax error.

The Maintenance utility returns this message when the syntax of the command you entered is incorrect. Verify the syntax before reentering the command.

BUTIL–6.10–11: The command line requires the index file.

If you specify the BUTIL –INDEX or -SAVE command (modified by the Y parameter) to the Btrieve Maintenance utility, you must specify the full pathname of an external index file.

BUTIL–6.10–12: The command line requires the key number.

If you specify the DROP command or the SAVE command (modified by the N parameter) to the Btrieve Maintenance utility, you must specify the key number of the key you want to drop or by which you want to save the Btrieve file.

BUTIL–6.10–13: The key size for key of type *xxxx* is invalid.

The Maintenance utility returns this message when you specify an invalid key size for the key type. In a description file, the specified value of the Key Length element for a particular key is incorrect. Make sure that the value of each Key Length element is appropriate for the matching Key Type element.

BUTIL–6.10–14: The key type is invalid.

The Maintenance utility returns this message when you specify an invalid key type. In a description file, the type specified for one of the keys is invalid. Make sure that all the Key Type elements in the file are assigned a valid type.

BUTIL–6.10–15: The key segment descriptor value *nn* is invalid for a manual or null key.

The Maintenance utility returns this message when you specify an invalid value for the key segment descriptor. The value for the key segment descriptor can be any hexadecimal value from 00 to FF. An example of an invalid value is GL.

BUTIL–6.10–16: BUTIL could not open the description file.

The Maintenance utility cannot open the description file. Before attempting to reenter the CREATE, INDEX, or SINDEX commands, make sure that the file exists and that you specify the correct full pathname.

BUTIL–6.10–18: An error occurred during access of the sequential file.

The Maintenance utility returns this message when an error occurs during access of a sequential file. Check to see if the Btrieve source file is valid.

BUTIL–6.10–19: BUTIL could not open the alternate collating sequence file.

The Maintenance utility cannot open the alternate collating sequence file that you specified in a description file. Make sure the Alternate Collating Sequence Filename element in the description file is assigned a valid pathname.

BUTIL–6.10–20: An error occurred during access of the alternate collating sequence file.

An error occurred when the Maintenance utility accessed the alternate collating sequence file. Make sure the information in the alternate collating sequence file is formatted correctly.

BUTIL–6.10–21: The file version is earlier than 6.0.

The Maintenance utility returns this message when the RECOVER command cannot recover data from a Btrieve 5.*x* file, or the SALVAGE command cannot repair or salvage a Btrieve 5.*x* file.

BUTIL–6.10–23: The /D parameter specified to the Requester was too small for BUTIL to receive the entire record. BUTIL is writing only *nn* bytes.

The Maintenance utility is writing only as many bytes as the value of the /D option allows. If you want the utility to write all the bytes in the record, specify a value for the /D option that is at least as large as the affected record.

BUTIL–6.10–25: The /D parameter specified to BUTIL was too small for BUTIL to receive any part of the record.

The Maintenance utility returns this message when you specify an invalid value for the /D option. Return to the Setup utility and increase the value specified for the Largest Record Size configuration option.

BUTIL–6.10–26: The data buffer is too small to hold any part of the record.

Btrieve cannot return any data in the data buffer because the data buffer is too small to hold it. Return to the Setup utility and increase the value specified for the Largest Record Size configuration option correctly in the description file.

BUTIL–6.10–27: An error occurred during the access of the variable page. BUTIL is writing the obtainable portion of the variable page.

The Maintenance utility returns this message when an error occurs during the recovery of a file with variable-length records. The file has been corrupted.

BUTIL–6.10–30: The key position cannot exceed the record length.

The Maintenance utility returns this message when the range of the key position you specified is invalid. The key position you specify on a Btrieve call must be within the range of the record's length. For example, for a record that is 100 bytes long, a key position of 50 is within the correct range. However, a key position of 150 is not.

BUTIL–6.10–31: The key position plus key length cannot exceed the record length.

The Maintenance utility returns this message when the range of the key position you specified is invalid. The key position of a key plus its length cannot be larger than the record length. Verify that the key is defined so that its position plus its length does not exceed the record length.

BUTIL–6.10–32: The key length must be an even number for key type *xxxx*.

The Maintenance utility returns this message when you specified an invalid key length for the key type. Some key types must contain an even number of bytes. Respecify the Key Length element correctly

BUTIL–6.10–36: The page size must be a multiple of 512, from 512 to 4,096.

The Maintenance utility returns this message if the page size you specified is not a multiple of 512, from 512 to 4,096. Specify an appropriate page size.

NetWare Btrieve specifies that 6.0 and 6.1*x* files with a page size of 512 can be no larger than 2 gigabytes in size, but does not enforce this limit. Therefore, when a file grows beyond the 512 page-size limit, NetWare Btrieve returns Status Code 132 (The file is full).

BUTIL–6.10–37: The record length cannot exceed the page size.

The Maintenance utility returns this message if the record length you specified is invalid. In the description file, the record length you specified for the Record Length element is larger than the page size you specified for the Page Size element. Specify a record length that is smaller than the page size or increase the page size.

BUTIL–6.10–38: The record length must be at least 4 and no greater than 4,096.

The Maintenance utility returns this message if the record length you specified is invalid. Specify a record length between 4 and 4,096 (inclusive) for Btrieve 5.*x*, or between 4 and 4,088 for Btrieve 6.*x*.

BUTIL–6.10–41: The alternate collating sequence cannot be found.

The Maintenance utility cannot find the alternate collating sequence file you specified in the definition file. Verify that the alternate collating sequence file exists and that the name is correct in the definition file.

BUTIL–6.10–43: The file exists, but the Replace option was not specified.

The Maintenance utility did not create a file when you specified the BUTIL – CREATE command because the file already exists. To recreate this file, specify the Replace Existing File element in the description file as y.

BUTIL–6.10–44: The file access error *nn* occurred for file *xxxx*.

The Maintenance utility returns the appropriate status code and filename for a file on which a file access error occurred during the beginning or end of continuous operation. The corrective measure depends on the status code received. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BUTIL–6.10–45: The number of duplicate keys must be between 1 and 119.

The number of duplicate keys you specified is invalid. Check the value specified for the Duplicate Key element in the description file.

BUTIL–6.10–47: BUTIL cannot open the command file.

The Maintenance utility returns this message when it cannot open the specified command file. Make sure the command file exists and that you specified the command file location and filename correctly.

BUTIL–6.10–48: The command file is empty.

The Maintenance utility returns this message if the command file you specified does not contain characters. Specify the desired commands in the command file before attempting to use the command file again. In addition, make sure you specified the correct command filename.

BUTIL–6.10–49: The command file exceeds 1,000 bytes.

A command file cannot contain more than 1000 bytes. Verify that the command file adheres to this requirement.

BUTIL–6.10–50: An internal error caused BUTIL to terminate.

The Maintenance utility detected an internal diagnostic error that caused it to terminate.

BUTIL–6.10–52: Btrieve cannot be stopped when Scalable SQL is loaded.

The Maintenance utility returns this message when you attempt to unload Btrieve while Scalable SQL is loaded. Unload Scalable SQL before attempting to unload Btrieve again.

BUTIL–6.10–53: Btrieve error *nn* occurred for file or command *xxxx*.

The Maintenance utility returns a status code related to a particular file or command. The corrective measure depends on the status code received. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BUTIL–6.10–56: BUTIL is processing source file record *nn*.

This is an informational message from the Maintenance utility. No action is required.

BUTIL–6.10–57: BUTIL has copied *nn* records so far.

The Maintenance utility copied the stated number of records since you issued the BUTIL –COPY command. After you receive this message, the command is still executing.

BUTIL–6.10–58: BUTIL copied *nn* records.

The Maintenance utility copied the stated number of records after you issued the BUTIL –COPY command. This is the total number of records BUTIL copied while the command executed.

BUTIL–6.10–59: BUTIL read *nn* records.

This is an informational message from the Maintenance utility. No action is required. When you issue a BUTIL –RECOVER command, the utility reads records in physical order from the specified file. When you issue a BUTIL –SAVE command, the utility reads records from the specified file using an index path.

BUTIL–6.10–60: The end of the file occurred while BUTIL was expecting keyword *xxxx* on key segment descriptor *nn*.

While the Maintenance utility was creating a file, it found a syntax error in the description file. Check the syntax of the description file.

BUTIL–6.10–61: The end of the file occurred while BUTIL was expecting keyword *xxxx*.

While the Maintenance utility was creating a file, it found a syntax error in the description file. Check the syntax of the description file.

BUTIL–6.10–62: BUTIL was expecting keyword *xxxx* on key segment descriptor *nn*.

While the Maintenance utility was creating a file, it found a syntax error in the description file. Check the syntax of the description file.

BUTIL–6.10–63: BUTIL was expecting keyword *xxxx*.

While the Maintenance utility was creating a file, it found a syntax error in the description file. Check the syntax of the description file.

BUTIL–6.10–64: *nn* records have been indexed.

This is an informational message from the utility. The utility displays the number of records that were indexed since you issued the BUTIL –INDEX command. No action is required.

BUTIL–6.10–65: BUTIL has loaded no records.

The Maintenance utility did not load any records after you specified the BUTIL –LOAD command. Verify that you specified the command correctly and that the input file is in the correct format.

BUTIL–6.10–66: BUTIL has loaded *nn* records so far.

The Maintenance utility loaded the stated number of records since you issued the BUTIL –LOAD command. When you receive this message, the command is still executing.

BUTIL–6.10–67: BUTIL is processing sequential record *nn*.

This is an informational message from the Maintenance utility. No action is required.

BUTIL–6.10–68: BUTIL has loaded *nn* records.

The Maintenance utility loaded the stated number of records after you issued the BUTIL –LOAD command. This is the total number of records BUTIL loaded while the command executed.

BUTIL–6.10–69: BUTIL has read *nn* sequential records.

This is an informational message from the Maintenance utility. No action is required.

BUTIL–6.10–70: The Btrieve error *nn* occurred on closing a file.

The Maintenance utility returns this status code while closing a file. The corrective measure depends on the status code received. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix.

BUTIL–6.10–71: BUTIL has recovered *nn* records so far.

The Maintenance utility recovered the stated number of records since you issued the BUTIL –RECOVER command. After you receive this message, the command is still executing.

BUTIL–6.10–72: BUTIL has recovered *nn* records.

The Maintenance utility recovered the stated number of records after you issued the BUTIL –RECOVER command. This is the total number of records BUTIL recovered while the command executed.

BUTIL–6.10–73: BUTIL is scanning the file.

The Maintenance utility is scanning a file while performing a BUTIL –SALVAGE command. No action is required.

BUTIL–6.10–74: Btrieve error *nn* was returned for the Stop Command.

The Maintenance utility returns this status code after the BUTIL -STOP command was issued. The corrective measure depends on the status code received. Consult the screen message to get the value for *nn*, which corresponds to a numeric status code listed in the section “Status Codes” in this appendix. This message applies only to the DOS environment.

BUTIL–6.10–76: When BUTIL wrote the Page Allocation Table at page #*nn*, an error occurred.

The Maintenance utility returns this message while salvaging a file if the file is corrupted or when a hardware error occurs.

BUTIL–6.10–77: When BUTIL wrote a mirror copy of the Page Allocation Table at page #*nn*, an error occurred.

The Maintenance utility returns this message while salvaging a file if the file is corrupted or when a hardware error occurs.

BUTIL–6.10–82: The file format is incorrect.

The Maintenance utility returns this message when the format for the file is prior to Btrieve 6.*x*. Check the format of the file.

BUTIL–6.10–83: BUTIL is checking Page Allocation Table page #*nn* of #*nn*.

The Maintenance utility returns this message while performing a BUTIL –SALVAGE. No action is required.

BUTIL–6.10–84: The Page Allocation Table entry *nn* on page *nn* at offset *nn* points to an invalid page.

The Maintenance utility returns this message when it encounters file corruption while performing a BUTIL –SALVAGE. You may lose some data. Check the salvaged file.

BUTIL–6.10–86: The file’s Page Allocation Table appears damaged.

The Maintenance utility returns this message while performing a BUTIL –SALVAGE. You may lose some data. Check the salvaged file.

BUTIL–6.10–90: BUTIL could not allocate enough memory.

The Maintenance utility cannot continue with the BUTIL –SALVAGE command due to inadequate memory at the server. Free some memory at the server by unloading unused NLMs.

BUTIL–6.10–91: BUTIL could not determine the size of the file.

The Maintenance utility cannot continue with the BUTIL –SALVAGE command. The file cannot be salvaged. Try to recover the file using the BUTIL –RECOVER command.

BUTIL–6.10–92: The user terminated BUTIL.

The Maintenance utility returns this message to indicate that the user entered 0 to quit when prompted by the utility.

BUTIL–6.10–93: BUTIL has saved *nn* records so far.

The Maintenance utility saved the stated number of records since you issued the BUTIL –SAVE command. When you receive this message, the command is still executing.

BUTIL–6.10–94: BUTIL has saved *nn* records.

The Maintenance utility saved the stated number of records after you issued the BUTIL –SAVE command. This is the total number of records BUTIL saved while the command executed.

BUTIL–6.10–128: The Btrieve Version is *n.n*.

The Maintenance utility returns the version of Btrieve that you are currently running.

BUTIL–6.10–131: BUTIL was unable to create or open the sequential file.

The Maintenance utility returns this message when it is unable to create or open the specified file. Check the sequential file to make sure it exists and has the read-only attribute set.

BUTIL–6.10–132: The disk volume is full.

The Maintenance utility returns this message when the disk volume is full. You must have more disk space to create or enlarge any Btrieve files.

BUTIL–6.10–134: BUTIL was unable to create or open the new file.

The Maintenance utility returns this message when it is unable to create or open the specified file. Check the file specified for the BUTIL –SAVE, –SALVAGE, or –RECOVER command. The file may already exist.

BUTIL–6.10–136: BUTIL was unable to write the new backup file.

The Maintenance utility returns this message when it is unable to write the new backup file. Verify that you specified the correct path and filename for the backup file. Also, make sure you have enough disk space for the file to be written.

BUTIL–6.10–152: There was an error opening file *xxxx*.

While performing the BUTIL –SALVAGE command, the Maintenance utility could not open the file. Check the Btrieve file attributes, path, and filename.

BUTIL–6.10–155: BUTIL cannot open the Btrieve file *xxxx*.

The Maintenance utility returns this message when it cannot open the specified file. Check the path, filename, and file attributes.

Btrieve Setup Utility for MS Windows

This section lists and describes the messages that the Setup utility for the MS Windows environment can send.

DBSETUP: NOVDB.INI contains invalid Btrieve parameters. The default values will be used for all parameters.

The Setup utility found invalid Btrieve for MS Windows configuration option settings in the initialization file NOVDB.INI. The Setup utility will use the default values for all Btrieve for MS Windows configuration options. For information about the configuration options and their default values, refer to the *Btrieve for MS Windows Installation and Operation* manual.

The NOVDB.INI file is a text file in your MS Windows directory and contains configuration option settings for Btrieve for MS Windows. For more information about the NOVDB.INI file and its required format, refer to the NOVDB.WRI file.

DBSETUP: Either Local or Remote must be checked. The default is Local.

In the Setup utility, you must specify either Local engine access or Remote engine access, or both. To continue, specify whether your Btrieve engine is local (client-based), remote (server-based), or both.

DBSETUP: The filename *filename* is incorrect.

In the Setup utility, you specified a filename that does not exist or is invalid. Ensure that the file exists and that the filename you specified is valid.

DBSETUP: The directory path *path* is incorrect.

In the Setup utility, you specified a directory path that does not exist or is invalid. Ensure that the path exists and that the pathname you specified is valid.

DBSETUP: The specified value is out of range. The valid range is *n* through *x*.

In the Setup utility, you specified a configuration option value that is out of the range of acceptable settings. Enter a value in the specified range.

For information about the ranges for each configuration option, refer to the *Installation and Operation* manual for your client-based environment.

DBSETUP: The utility cannot create a temporary file and therefore cannot complete the conversion.

The Setup utility is unable to create a temporary file for use in converting Btrieve files. Check for disk full, disk write-protected, disk offline, or invalid access rights conditions.

DBSETUP: The value specified for Files To Be Converted is invalid.

In attempting to convert Btrieve files using the Setup utility, you specified a filename that does not exist or is invalid. Ensure that the file exists and that the filename you specified for conversion is valid.

DBSETUP: The value specified for Output Directory is invalid.

In attempting to convert Btrieve files using the Setup utility, you specified a directory path that does not exist or is invalid. Ensure that the path exists and that the pathname you specified is valid.

DBSETUP: The values specified for Files to Be Converted and Output Directory are invalid.

In attempting to convert Btrieve files using the Setup utility, you specified both a filename and a pathname that do not exist or are invalid. Ensure that the file and path exist and that the filename and pathname you specified are valid.

DBSETUP: The log file `filename` is too large to display.

During conversion of Btrieve files, the Setup utility created a log file that is too large to display in the utility. The log file, called REBUILD.LOG, is a text file you can view using an ASCII text editor.

DBSETUP: Insufficient memory is available to display `filename`.

The Setup utility cannot allocate enough memory to display the specified log file. Ensure that your workstation meets the system requirements discussed in the *Installation and Operation* manual for your client-based environment.

DBSETUP: An I/O error occurred when the utility attempted to display `filename`.

The Setup utility encountered an I/O error when it attempted to display the specified log file. Ensure that your workstation meets the system requirements discussed in the *Installation and Operation* manual for your client-based environment. Also, check for disk full, disk write-protected, disk offline, or invalid access rights conditions.

Roll Forward Utility for OS/2

This section lists and describes the messages that the Roll Forward utility for the OS/2 environment can send.

PBROLL–6.10–1: Window creation failed.

The underlying cause of this condition is insufficient free memory. To complete the operation, you must free some of the memory currently allocated or terminate some of the active applications.

PBROLL–6.10–2: Window positioning failed.

The OS/2 Roll Forward utility returns this message when memory has been corrupted. Try turning your computer off and on to reinitialize the memory.

PBROLL–6.10–3: Cannot open the following file: *`filename`*.

The OS/2 Roll Forward utility cannot open the specified file (Btrieve file, log file, or listing file). Make sure that the specified file exists and then specify the correct, full pathname.

PBROLL–6.10–4: An error occurred during input or output for the following file: *`filename`*.

Btrieve either cannot read from or cannot write to the disk. Make sure that the file attributes are set correctly. You may also try running server diagnostics or checking the disk subsystem.

PBROLL–6.10–5: The following file ended unexpectedly: *`filename`*.

The OS/2 Roll Forward utility returns this message when the log file for the specified Btrieve file is corrupt. You cannot use this log file.

PBROLL–6.10–6: There is no header information in the following file: *`filename`*.

The OS/2 Roll Forward utility cannot find the log file header information in the specified log file because the log file is corrupt. You cannot use this log file.

PBROLL–6.10–7: There is incorrect header syntax in the following file: *`filename`*.

The OS/2 Roll Forward utility returns this message when the specified log file is corrupt and contains incorrect syntax in the log file header. You cannot use this log file.

PBROLL–6.10–8: The following file was not found in BLOG.CFG: *`filename`*.

The OS/2 Roll Forward utility returns this message when it cannot find the specified file in the log configuration file. Edit the BLOG.CFG file to add the filename specified. For more information about the BLOG.CFG file, refer to the section “Creating the Log Configuration File” in Chapter 5.

PBROLL–6.10–9: The length of the record exceeds the current data buffer size for the following file: *`filename`*.

The record size of the specified Btrieve file exceeds the data length you specified in the Options dialog box. Change the data length accordingly.

PBROLL–6.10–10: Btrieve error: Operation: *nn* Status Code: *nn*

Btrieve returns the specified Btrieve operation and status codes to the OS/2 Roll Forward utility. The specified Btrieve operation was unsuccessful. Consult the screen message to get the value of Status Code *nn*, which corresponds to a numeric status code listed in the “Status Codes” section of this appendix. For more information about the Btrieve operation *nn*, refer to the *Btrieve Programmer's Manual* for your environment.

PBROLL–6.10–11: The following unknown error occurred: *nn*.

The OS/2 Roll Forward utility detected an internal diagnostic error.

PBROLL–6.10–12: There was an invalid owner name.

The OS/2 Roll Forward utility returns this message when the specified owner name is invalid. Specify a valid owner name and start the roll forward process again. The maximum length for an owner name is 8 bytes. Enter a valid name in the Owner text box.

PBROLL–6.10–13: Unable to add the item to the queue.

The OS/2 Roll Forward utility works on a queued-job basis. The maximum number of files allowed in the queue is 32. That is, the utility cannot roll forward more than 32 files at one time.

If you want to roll forward all the files on a specified volume, edit the log configuration file, BLOG.CFG, as needed to ensure that no more than 32 files are specified. If you want to roll forward individual files, make sure that you do not specify more than 32 files at one time.

PBROLL–6.10–14: There is not enough memory available.

The workstation does not have enough memory for the OS/2 Roll Forward utility to complete the current operation. Free some memory on the workstation and restart the roll forward process.

PBROLL–6.10–15: The data length value is invalid. It must be between 1 and 64.

The OS/2 Roll Forward utility returns this message when the specified data length is invalid. The Data Length option in the Options dialog box specifies the number of kilobytes allocated for the data buffer that the utility uses to process the logged entries. The value specified for the Data Length option should be at least as large as the largest record to be rolled forward. Specify an appropriate value for the Data Length option in the Options dialog box.

PBROLL–6.10–16: Could not allocate the data buffer.

The OS/2 Roll Forward utility returns this message when the system is low on memory. Free some system memory or specify a smaller data buffer, and run the utility again.

PBROLL–6.10–17: Could not open the list file.

The OS/2 Roll Forward utility returns this message when the pathname you entered for the new list file (when the disk ran out of space) is invalid. Specify a valid pathname for the new list file.

PBROLL–6.10–18: ABORT—The data files may be left in an inconsistent state. Please confirm abort.

The OS/2 Roll Forward utility returns this message when you select the Abort option during the roll forward process. Selecting Abort prevents the utility from applying the rest of the logged operations to the backup copy of the corresponding Btrieve file. Confirming the abort can leave the Btrieve file that is currently being rolled forward in an inconsistent state (that is, only partially updated). Select Yes to confirm you want to abort or No to cancel the abort and continue the roll forward process.

PBROLL–6.10–19: This utility could not open NOVDB.INI.

The OS/2 Roll Forward utility returns this message when it cannot open the initialization file, NOVDB.INI. Check the file attributes of the NOVDB.INI file to ensure that you have access rights to it. This file should be in the directory you selected when choosing the options for the utility.

PBROLL–6.10–20: This item or volume is already in the queue.

The OS/2 Roll Forward utility returns this message when the item or volume you entered in the queue has already been entered. Enter a different item or volume.

PBROLL–6.10–21: The listing length exceeds the buffer length.

The OS/2 Roll Forward utility returns this message when the specified listing length is invalid. In the Options dialog box, the Data to List option exceeds the value you entered for the Data Length option.

The Data to List option specifies the length of the data buffer that will be printed in the list file for each operation that is rolled forward. The Data Length option specifies the number of kilobytes allocated for the data buffer that the utility uses to process the logged entries. Make sure the value you enter for the Data to List option does not exceed the value you entered for the Data Length option.

PBROLL–6.10–22: The listing length is invalid.

The OS/2 Roll Forward utility returns this message when the specified listing length is invalid. You entered an invalid value for the Data to List or the Key to List option in the Options dialog box.

The Data to List option specifies the length of the data buffer that will be printed in the list file for each operation that is rolled forward. Similarly, the Key to List option specifies the length of the key buffer that will be printed in the list file for each operation that is rolled forward.

In the options dialog box, make sure the value you enter for the Data to List option does not exceed the value you entered for the Data Length option and the Key to List option does not exceed 255 bytes.

PBROLL–6.10–23: The key listing length cannot exceed 255.

The OS/2 Roll Forward utility returns this message when the specified key listing length is invalid. The value you entered for the Key to List option in the Options dialog box exceeds 255 bytes. The Key to List option specifies the length of the key buffer that will be printed in the list file for each operation that is rolled forward. Make sure the value you enter for the Key to List option does not exceed 255 bytes.

PBROLL–6.10–24: Only one instance is allowed to run.

You tried to run the OS/2 Roll Forward utility, but it is already running.

Btrieve Function Executor Utility

This section lists and describes the messages that the Btrieve Function Executor utility for the MS Windows environment can send.

WBEXEC-6.10-1 Insufficient memory is available to display entire data buffer. Data is truncated to *n* bytes.

The Function Executor was unable to allocate enough memory to return the entire data buffer. Consult your MS Windows documentation for information about freeing available memory while running applications. Also, ensure that your workstation meets the system requirements described in the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-2 The utility could not create the Btrieve Function Executor screen.

A critical error occurred that prevents the Function Executor from displaying its screen. Restart MS Windows and open the Function Executor again.

WBEXEC-6.10-3 The specified value exceeds the maximum for the current operation. The maximum is 32,767.

In the Function Executor, you specified a value that is outside the range of acceptable settings. Enter a value of 32,767 or less. For more information about the acceptable values you can enter using the Function Executor, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-4 No file is currently opened for this operation.

In the Function Executor, you attempted to perform an operation that requires you to open a file first. To open a file, select the Open (0) operation in the List box, enter a filename in the Key Buffer text box, and choose the Execute button. For more information about how to open a file using the Function Executor, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-5 A global memory locking error occurred.

The Function Executor is unable to allocate the memory required to perform the function you requested. Consult your MS Windows documentation for information about freeing available memory while running applications.

WBEXEC-6.10-6 The specified value exceeds the maximum for the Buffer Length. The maximum is 65,535.

In the Function Executor, you specified a value that is outside the range of acceptable values for the Buffer Length box in the Data group. Enter a value of 65,535 or less. For more information about the Buffer Length box, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-7 The current operation requires a Buffer Length of at least 4.

In the Function Executor, the operation you requested requires a Buffer Length of at least 4. Enter a value of 4 or more in the Buffer Length box of the Data group. For more information about the Buffer Length box, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-8 The specified value exceeds the maximum for the Key Number. The maximum is 32,767.

In the Function Executor, you specified a value that is outside the range of acceptable settings for the Number box in the Key group. Enter a value of 32,767 or less. For more information about the Number box, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-9 This operation requires a key number.

In the Function Executor, you attempted to perform an operation that requires you to specify a key number. For information about the operations that require a key number, refer to the *Btrieve for MS Windows Programmer's Manual*. For information about using the Function Executor, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBEXEC-6.10-10 The maximum number of files(255) are open. Close one or more files before attempting to open another file.

In the Function Executor, you reached the maximum number of files that can be open on your workstation at one time.

WBEXEC-6.10-11 Insufficient memory is available for resizing the data buffer.

The Function Executor is unable to allocate the memory required to resize the data buffer. Consult your MS Windows documentation for information about freeing available memory while running applications.

WBEXEC-6.10-12 Insufficient memory is available for performing the current task.

The Function Executor is unable to allocate the memory required to perform the function you requested. Consult your MS Windows documentation for information about freeing available memory while running applications.

WBEXEC-6.10-13 Either a file is open or a transaction is active. Do you want to reset the file or transaction?

You attempted to exit the Function Executor while a file is still open or a transaction is still active. If you choose Yes, the Function Executor closes all open files, releases any locks, aborts all active transactions, and exits. If you choose No, the Function Executor and its files remain open.

Btrieve File Manager Utility

This section lists and describes the messages that the Btrieve File Manager utility for the MS Windows environment can send.

WBMANAGE-6.10-1: The maximum number of key segments has been reached. The maximum is *n*.

In the File Manager, you reached the maximum number of key segments for the current file. The number of key segments allowed is based on the file's page size. For more information about key segments and page sizes, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-2: The Btrieve file *`filename`* already exists. The total number of records is *n*. Continuing will create a new file and destroy all existing records. Do you want to continue?

You specified a filename that already exists. If you choose Yes, the File Manager creates a new file with the same filename. All existing records are destroyed.

WBMANAGE-6.10-3: The *`type`* file *`filename`* was created successfully.

The File Manager successfully created the specified file.

WBMANAGE-6.10-4: The editor contents have changed. Do you want to create a new file before continuing?

Information has changed in the File Manager's editor. You can create a new file or change the information in the existing file.

WBMANAGE-6.10-5: The *`type`* file *`filename`* was opened successfully.

The File Manager successfully opened the specified file.

WBMANAGE-6.10-6: The key *nn* was created successfully.

The File Manager successfully added the specified key to the Btrieve file.

WBMANAGE-6.10-7: The key *nn* was dropped successfully.

The File Manager successfully removed the specified key from the Btrieve file.

WBMANAGE-6.10-8: The ACS file *`filename`* for key *x*, segment *y* could not be found.

Either you specified a nonexistent alternate collating sequence (ACS) file in the ACS File box or you opened a description file that specifies a nonexistent ACS file in the `name=filename` element. Specify an existing ACS file.

WBMANAGE-6.10-9: The ACS file *`filename`* is invalid for key *x*, segment *y*.

The filename for the specified alternate collating sequence (ACS) file is invalid. Ensure that the file meets the specifications for ACS files, as discussed in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-10: The non-Btrieve file *`filename`* already exists. Do you want to replace it?

You specified a filename that already exists and the existing file is not a Btrieve file. If you choose Yes, the File Manager creates a new file with the same filename. All contents in the existing file are destroyed.

WBMANAGE-6.10-11: The ACS list for key number *`n`* has an invalid value.

The contents of the alternate collating sequence (ACS) file are invalid. Ensure that the file meets the specifications for ACS files, as discussed in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-12: The owner was set successfully.

The File Manager successfully set the owner name for the Btrieve file.

WBMANAGE-6.10-13: The owner was cleared successfully.

The File Manager successfully cleared the owner name for the Btrieve file.

WBMANAGE-6.10-14: You must specify a source file.

In the File Manager, you omitted a source filename. Specify a source file.

WBMANAGE-6.10-15: You must specify a destination file.

In the File Manager, you omitted a destination filename. Specify a destination file.

WBMANAGE-6.10-16: You must specify an external index file.

In the File Manager, you omitted an external index filename. Specify an external index file.

WBMANAGE-6.10-17: File *`filename`* does not exist.

In the File Manager, you specified a file that does not exist. Specify an existing file.

WBMANAGE-6.10-18: File *filename* exists. Do you want to replace it?

You specified a filename that already exists. If you choose Yes, the File Manager creates a new file with the same filename. All contents in the existing file are destroyed.

WBMANAGE-6.10-19: You must specify a Btrieve file.

In the File Manager, you specified a file that is not a Btrieve file. Specify a file that is readable by Btrieve.

WBMANAGE-6.10-20: Preallocation pages must be between *x* and *y*.

In the File Manager, you specified a value for preallocation pages that is outside the range of acceptable values. Specify an acceptable value.

WBMANAGE-6.10-21: The record length must be between *x* and *y*.

In the File Manager, you specified a record length that is outside the range of acceptable values. Specify an acceptable value.

WBMANAGE-6.10-22: Available duplicate pointers must be between *x* and *y*.

In the File Manager, you specified a value for available duplicate pointers that is outside the range of acceptable values. Specify an acceptable value.

WBMANAGE-6.10-23: Btrieve returned Status Code *nn* while the owner for file *filename* was being set.

Btrieve returned the specified status code when you attempted to set the owner for the specified file using the File Manager. For more information about status code *nn*, refer to the status code description in the “Status Codes” section of this appendix.

WBMANAGE-6.10-25: An error occurred while the file *filename* was being saved. *Message.*

The File Manager encountered an error while saving the specified file. The subsequent message provides more information about the nature of the error.

WBMANAGE-6.10-26: The message file `filename` could not be opened. Status Code `nn`.

Btrieve returned the specified status code when you attempted to open the specified file using the File Manager. For more information about status code `nn`, refer to the status code description in the “Status Codes” section of this appendix.

WBMANAGE-6.10-27: An error occurred but the error message could not be retrieved.

The File Manager encountered an error, but was unable to display an error message. Ensure that you installed Btrieve according to the instructions in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-28: The file `filename` could not be opened as a Btrieve file. `Message`. Also, the file could not be opened as a description file. `Message.`

The File Manager encountered an error while opening the specified file. The subsequent message provides more information about the nature of the error. Ensure that the file is either a Btrieve file or a description file.

WBMANAGE-6.10-30: An error was found in the description file. Btrieve does not support specifying data=n and variable=y and is ignoring variable=y.

The description file contains both a negative Include Data element (`data=n`) and a positive Variable-Length Records element (`variable=y`). Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both elements, the File Manager is ignoring the `variable` element and treating the file as if it had fixed-length records.

For more information about description file elements, refer to the *`Btrieve for MS Windows Installation and Operation`* manual.

WBMANAGE-6.10-31: An error was found in the description file. Btrieve does not support specifying data=n and truncate=y and is ignoring truncate=y.

The description file contains both a negative Include Data element (`data=n`) and a positive Blank Truncation element (`truncate=y`). Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both elements, the File Manager is ignoring the `truncate` element and reading the file without blank truncation.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-32: An error was found in the description file. Btrieve does not support specifying data=n and fthreshold=n and is ignoring fthreshold=n.

The description file contains both a negative Include Data element (`data=n`) and a Free Space Threshold element (`fthreshold= 10|20|30`). Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both elements, the File Manager is ignoring the `fthreshold` element and treating the file as if it had no free space threshold.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-33: An error was found in the description file. Btrieve does not support specifying data=n and vats=y and is ignoring vats=y.

The description file contains both a negative Include Data element (`data=n`) and a positive Variable-Tail Allocation Tables element (`vats=y`). Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both elements, the File Manager is ignoring the `vats` element and treating the file as if it had no Variable-Tail Allocation Tables.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-34: An error was found in the description file. Btrieve does not support specifying data=n and compress=y and is ignoring compress=y.

The description file contains both a negative Include Data element (`data=n`) and a positive Data Compression element (`compress=y`). Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both elements, the File Manager is ignoring the `compress` element and treating the file as if it had no data compression.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-35: An error was found in the description file. The value for page=*nnnn* is invalid. *N* will be used for the page size.

The description file contains an invalid value for the Page Size element. The File Manager uses the specified page size instead. Valid page sizes are multiples of 512, up to 4,096.

For more information about the Page Size description file element, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-36: An error was found in the description file. The value for record=*nnnn* is invalid. *N* will be used for the record length.

The description file contains an invalid value for the Record Length element. The File Manager uses the specified record length instead. The data record length must be at least 4 bytes and must be large enough to contain all the keys defined for the file. The record length (including its duplicate key overhead and usage count overhead) plus 6 bytes must not exceed the file's page size.

For more information about the Record Length description file element, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-37: An error was found in the description file. The value for key=*nnn* is invalid. *N* will be used for the number of keys.

The description file contains an invalid value for the Key Count element. The File Manager uses the specified key count instead. The file's page size limits the amount of key segments a file can have. For example, files with a page size of 512 bytes can have up to 8 key segments, whereas files with a page size of 4,096 bytes can have up to 119 key segments.

For more information about the Key Count description file element, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-38: An error was found in the description file. Specifying compress=y and truncate=y is not supported. Truncate=y will be ignored.

The description file contains both a positive Data Compression element (`compress=y`) and a positive Blank Truncation element (`truncate=y`). Because Btrieve does not support specifying both elements, the File Manager is ignoring the `truncate` element and treating the file as if it had no blank truncation.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-39: An error was found in the description file for key *x*, segment *y*. The value for position=*nnn* is invalid. *N* will be used for the key position.

The description file contains an invalid value for the Key Position element. The File Manager uses the specified key position instead.

For more information about the Key Position description file element, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-40: An error was found in the description file for key *x*, segment *y*. Specifying caseinsensitive=y and an ACS name is not supported. The ACS name will be ignored.

The description file contains both a positive Case-Insensitive Key element (`caseinsensitive=y`) and an Alternate Collating Sequence Filename element (`name=`*`filename`*). Because Btrieve does not support specifying both elements, the File Manager is ignoring the `name` element and treating the file as if it had no alternate collating sequence file.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-41: An error was found in the description file for key *x*, segment *y*. The value for length=*nnn* is invalid. *N* will be used for the key length.

The description file contains an invalid value for the Key Length element. The File Manager uses the specified key length instead. The value you specify for the key length cannot exceed the limit dictated by the key type.

For specific information about the key types and lengths, refer to the *Btrieve for MS Windows Programmer's Manual*. For more information about the Key Length description file element, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-42: An error was found in the description file. The value specified for keynum=*x* is not unique and ascending. keynum=*y* will be used.

The description file contains a Key Number element that has one or both of the following errors:

- The element attempts to define a key already defined in the file.
- The element attempts to define the key in a non-ascending order. That is, you can define keys nonsequentially (for example, key numbers 1, 3, and 5), but you cannot define keys in a non-ascending order (for example, key numbers 1, 5, and 3).

For more information about description files, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-43: An error was found in the description file. If you specify data=n, you must specify exactly one key definition. Btrieve is ignoring data=n.

The description file contains both a negative Include Data element (`data=n`) and more than one key definition. Specifying `data=n` creates a key-only file. Because Btrieve does not support specifying both, the File Manager is ignoring the `data` element and treating the file as if it were not a key-only file.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-44: An error was found in the description file. Specifying an ACS name and a nonstring data type is not supported. The ACS name will be ignored.

The description file contains an Alternate Collating Sequence Filename element (`name=`*`filename`*) for a key that is not a string data type. Because Btrieve does not support alternate collating sequences for nonstring data types, the File Manager is ignoring the `name` element.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-45: An error was found in the description file. The specified physical record length, *n*, is too large for the specified page size, *x*. *Y* will be used for the page size.

The description file sets a record length that is too long for the file's page size. The File Manager uses the specified record length instead. The record length (including its duplicate key overhead and usage count overhead) plus 6 bytes must not exceed the file's page size.

WBMANAGE-6.10-46: An error was found in the description file. The sum of keys with linked duplicates and available linked keys must be less than *x* for a page size of *y*. *N* will be used for the available linked keys.

The description file sets a number of linked-duplicatable keys and available linked keys that exceeds the number allowed for the current page size. The File Manager uses the specified page size instead.

For more information about description file elements, refer to the *Btrieve for MS Windows Installation and Operation* manual. For information about linked-duplicatable keys, refer to the *Btrieve for MS Windows Programmer's Manual*.

WBMANAGE-6.10-47: An error was found in the description file. The specified page size, *x*, is too small to contain 8 values for key=*y*. *N* will be used for the page size.

In the File Manager, you specified a page size that is too small to contain eight key values. You must be able to fit 8 key values in each page. For more information about determining an optimum page size, refer to the *Btrieve for MS Windows Programmer's Manual*.

WBMANAGE-6.10-48: An error was found in the description file. The specified number of segments, *x*, is too large for the specified page size, *y*. *N* will be used for the page size.

The description file sets a number of key segments that is too large for the file's page size. The File Manager uses the specified page size instead. The file's page size limits the amount of key segments a file can have. For example, files with a page size of 512 bytes can have up to 8 key segments, whereas files with a page size of 4,096 bytes can have up to 119 key segments.

For more information about key segments and page sizes, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-49: The `attribute` specification for key *x*, segment *y* is inconsistent with the previous segment. The specification will be set to the value designated in key *x*, segment *n*.

The description files sets inconsistent key attributes for the specified key segment. The File Manager uses the attributes in the specified key segment instead.

The elements `duplicates`, `modifiable`, `repeatdup`, and (alternate collating sequence) `name` apply only to entire keys. That is, you cannot apply one of these attributes to a single key segment without applying them to the entire key. For example, if you create a segmented key and specify `modifiable=y` for one of the key segments, you must specify `modifiable=y` for every segment in that key.

In contrast, you can apply `caseinsensitive`, `descending`, and `null` to individual key segments without affecting the entire key. For example, you can create a case-insensitive key segment in an otherwise case-sensitive key.

WBMANAGE-6.10-50: The specified number of segments, *x*, is too large for the specified page size, *y*.

In the File Manager, you specified a number of key segments that is too large for the file's page size. The file's page size limits the amount of key segments a file can have. For example, files with a page size of 512 bytes can have up to 8 key segments, whereas files with a page size of 4,096 bytes can have up to 119 key segments.

For more information about key segments and page sizes, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-51: The sum of keys with linked duplicates and available linked keys must be less than *x* for a page size of *y*.

In the File Manager, you specified a total number of linked-duplicatable keys and available linked keys that exceeds the number allowed for the current page size. Either specify fewer linked-duplicatable keys or specify a larger page size.

For more information about linked-duplicatable keys, refer to the *Btrieve for MS Windows Programmer's Manual.*

WBMANAGE-6.10-52: The key position for key *x*, segment *y* is invalid. The key position must be less than the record length.

In the File Manager, you specified a key position that is too high for the file's record length. Specify a key position that does not exceed the record length.

For more information about specifying key positions, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-53: The key length for key *x*, segment *y* is invalid. The sum of the position and the length must be less than the record length.

In the File Manager, you specified a key position and length whose sum exceeds the file's record length. Specify a key position and length whose sum does not exceed the record length. For more information about specifying key positions and lengths, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-54: The specified page size, *x*, is too small to contain 8 values for key=*y*.

In the File Manager, you specified a page size that is too small to contain eight key values. You must be able to fit 8 key values in each page. For more information about determining an optimum page size, refer to the *Btrieve for MS Windows Programmer's Manual*.

WBMANAGE-6.10-55: The specified physical record length, *n*, is too large for the specified page size, *x*.

In the File Manager, you specified a record length that is too long for the file's page size. The record length (including its duplicate key overhead and usage count overhead) plus 6 bytes must not exceed the file's page size.

WBMANAGE-6.10-56: The maximum record length with data compression enabled is *n*.

In the File Manager, you specified a record length that exceeds the maximum record length allowed with data compression turned on. Either specify a smaller record length or turn data compression off.

WBMANAGE-6.10-57: The key length for key *x* is greater than the maximum Btrieve key length of *y*.

In the File Manager, you specified a key length that exceeds the maximum length Btrieve allows for that key. Specify a valid length for the key. For specific information about the key types and lengths, refer to the *Btrieve for MS Windows Programmer's Manual*.

WBMANAGE-6.10-58: If you specify the key only attribute, you must specify exactly one key definition.

In the File Manager, you turned the Key Only attribute on and attempted to specify more than one key definition. Key-only files can contain only one key. For more information about using the Key Only attribute, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-59: The key position must be between *x* and *y*.

In the File Manager, you must enter a key position between the specified values. The key position value cannot exceed the value you specified for the record length.

WBMANAGE-6.10-60: The key length must be between *x* and *y*.

In the File Manager, you must enter a key length between the specified values. The value you specify for the key length cannot exceed the limit dictated by the key type. For specific information about the key types and lengths, refer to the *Btrieve for MS Windows Programmer's Manual*.

WBMANAGE-6.10-61: You cannot save the file information when using the Key Only option unless you have specified a key.

In the File Manager, you turned the Key Only attribute on and attempted to save the file before specifying a key. Key-only files must have a single key. For more information about using the Key Only attribute, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-62: File *`filename`* already exists. Btrieve is not loaded and cannot determine whether the file is a Btrieve file. Do you want to replace the file?

In the File Manager, you attempted to save the contents of a file to another file that already exists. If you choose Yes, the File Manager replaces the existing file contents with the new file contents. For more information about the Save command, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-63: An initialization error occurred. Please make certain all components are installed.

The File Manager encountered an error while attempting to initialize. Ensure that you have installed Btrieve for MS Windows according to the instructions provided in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-64: The free memory available is insufficient.

The File Manager is unable to allocate the memory required to perform the function you requested. Consult your MS Windows documentation for information about freeing available memory while running applications.

WBMANAGE-6.10-65: The maximum key number, *n*, has been reached.

In the File Manager, you attempted to create more keys than Btrieve allows. Btrieve 6.*x* supports up to 119 key segments in files with a page size of 4,096 bytes. Versions of Btrieve earlier than 6.0 supported a maximum of 24 key segments.

WBMANAGE-6.10-66: The Create option is not available. No key specifications have been defined in the editor.

In the File Manager, you attempted to create an index but have not defined any keys. Use the editor to define a key, then create the index. For more information about using the File Manager, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-67: This file has a pre-version 6.0 format. These restrictions apply:
The Create Index operation will always use the next available key number.
The Drop Index operation will always renumber any remaining indexes.
Only supplemental indexes can be dropped.

In the File Manager, you are working with a file with a format previous to Btrieve 6.*x*. When you are working with pre-6.*x* files, the specified restrictions apply. If you are working with a 5.*x* file and want to convert it to 6.*x* format, refer to the discussion of the Setup utility in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-68: The key number is invalid. This index is a permanent index. For a file with a format earlier than version 6.0, you can drop only supplemental indexes.

In the File Manager, you attempted to drop a key with an index that is defined as permanent. Btrieve 6.*x* supports adding and dropping any index. In files with a format earlier than 6.0, you can add and drop only supplemental indexes (that is, only those indexes that are not originally produced during a Create operation).

If you are working with a 5.*x* file and want to convert it to 6.*x* format, refer to the discussion of the Setup utility in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-69: The file `filename` was created successfully. The file format is earlier than version 6.x. Only pre-6.x data is shown. Press the Show 6.x Data button to edit 6.x data.

The File Manager successfully created the specified file in pre-6.0 format. For more information about using the File Manager, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-70: The file `filename` was opened successfully. The file format is earlier than version 6.x. Only pre-6.x data is shown. Press the Show 6.x Data button to edit 6.x data.

The File Manager successfully opened the specified file. For more information about using the File Manager, refer to the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-71: The file `filename` could not be opened. The file does not exist.

The File Manager could not open the specified file because it does not exist. Specify an existing Btrieve file or description file.

WBMANAGE-6.10-72: The available free space is insufficient.

The File Manager was unable to allocate enough memory to perform the operation you requested. The File Manager allocated as much memory as possible. Consult your MS Windows documentation for information about freeing available memory while running applications.

Also, ensure that your workstation meets the system requirements described in the *Btrieve for MS Windows Installation and Operation* manual.

WBMANAGE-6.10-73: A critical error occurred. The available free memory is insufficient.

The File Manager encountered a memory error that prevents the utility from continuing. Exit MS Windows and restart the File Manager.

WBMANAGE-6.10-74: The background task could not be initialized.

The File Manager encountered an error while trying to start a background process. First, attempt to exit and restart the File Manager. If necessary, exit and restart MS Windows.

Btrieve Roll Forward Utility for MS Windows

This section lists and describes the messages that the Roll Forward utility for the MS Windows environment can send.

WBROLL–6.1–1: The utility could not open configuration file, BLOG.CFG.

The MS Windows Roll Forward utility cannot open the log configuration file, BLOG.CFG. Make sure that the file is in the system directory SYS:\SYSTEM. For more information about the BLOG.CFG file, refer to the section "Creating the Log Configuration File" in Chapter 5.

WBROLL–6.1–2: Do you want to continue the roll forward?

The MS Windows Roll Forward utility returns this message when one of the files cannot be rolled forward. You can continue rolling forward the other files, or you can stop the roll forward process. Select Yes to continue or No to stop.

Stopping the roll forward process prevents the Roll Forward utility from applying the rest of the logged operations to the backup copy of the corresponding Btrieve file. This can leave your Btrieve files in an inconsistent state (that is, only partially updated).

WBROLL–6.1–3: The utility cannot open the following file: *`filename`*.

The MS Windows Roll Forward utility cannot open the specified file (Btrieve file, log file, or listing file). Make sure that the specified file exists and then specify the correct, full pathname.

WBROLL–6.1–4: An error occurred during input or output for the following file: *`filename`*.

Btrieve either cannot read from or cannot write to the disk. Make sure that the file attributes are set correctly. You may also try running server diagnostics.

WBROLL–6.1–5: The following file ended unexpectedly: *`filename`*.

The MS Windows Roll Forward utility returns this message when the log file for the specified Btrieve file is corrupt. You cannot use this log file.

WBROLL–6.1–6: There is no header information in the following file: *`filename`*.

The MS Windows Roll Forward utility cannot find the log file header information in the specified log file because the log file is corrupt. You cannot use this log file.

WBROLL–6.1–7: The header syntax in the following file is incorrect: *`filename`*.

The MS Windows Roll Forward utility returns this message when the specified log file is corrupt and contains incorrect syntax in the log file header. You cannot use this log file.

WBROLL–6.1–8: The following file was not found in BLOG.CFG: *`filename`*.

The MS Windows Roll Forward utility returns this message when the specified file is not found in the log configuration file, BLOG.CFG. Edit the BLOG.CFG file to add the filename specified. For more information about the BLOG.CFG file, refer to the section “Creating the Log Configuration File” in Chapter 5.

WBROLL–6.1–9: The length of the record exceeds the current data buffer size for the following file: *`filename`*.

The MS Windows Roll Forward utility returns this message when the specified record length is invalid. The record size of the specified Btrieve file exceeds the data length you specified in the Options dialog box. Change the data length accordingly.

WBROLL–6.1–10: A Btrieve error occurred. Operation Code: *nn*, Status Code: *nn*

Btrieve returns the specified Btrieve operation and status codes to the MS Windows Roll Forward utility. The specified Btrieve operation was unsuccessful. Consult the screen message to get the value of Status Code *nn*, which corresponds to a numeric status code listed in the “Status Codes” section of this appendix. For more information about the Btrieve operation *nn*, refer to the *Btrieve Programmer's Manual*.

WBROLL–6.1–11: The following unknown error occurred: *nn.*

The MS Windows Roll Forward utility detected an internal diagnostic error.

WBROLL–6.1–12: There was an invalid owner name.

The MS Windows Roll Forward utility returns this message when the specified owner name is invalid. Specify a valid owner name and start the roll forward process again. The maximum length for an owner name is 8 bytes. Enter a valid name in the Owner text box.

WBROLL–6.1–13: The utility is unable to add the item to queue.

The MS Windows Roll Forward utility works on a queued-job basis. The maximum number of files allowed in the queue is 32. That is, the utility cannot roll forward more than 32 files at one time.

If you want to roll forward all the files on a specified volume, edit the log configuration file, BLOG.CFG, as needed to ensure that no more than 32 files are specified. If you want to roll forward individual files, make sure that you do not specify more than 32 files at one time.

WBROLL–6.1–14: Insufficient memory is available.

The workstation does not have enough memory for the MS Windows Roll Forward utility to complete the current operation. Free some memory on the workstation and restart the roll forward process.

WBROLL–6.1–15: The data length value is invalid. It must be between 1 and 64.

The MS Windows Roll Forward utility returns this message when the specified data length is invalid. The Data Length option in the Options dialog box specifies the number of kilobytes allocated for the data buffer that the utility uses to process the logged entries.

The value specified for the Data Length option should be at least as large as the largest record to be rolled forward. Specify an appropriate value for the Data Length option in the Options dialog box.

WBROLL–6.1–16: The utility could not allocate the data buffer.

The MS Windows Roll Forward utility returns this message when the system is low on memory. Free some system memory or specify a smaller data buffer, and run the utility again.

WBROLL–6.1–17: The utility could not open the list file.

The MS Windows Roll Forward utility returns this message when the pathname you entered for the new list file (when the disk ran out of space) is invalid. Specify a valid pathname for the new list file.

WBROLL–6.1–18: ABORT — The data files may be left in an inconsistent state. Do you want to abort the operation?

The MS Windows Roll Forward utility returns this message when you select the Abort option during the roll forward process. Selecting Abort prevents the Roll Forward utility from applying the rest of the logged operations to the backup copy of the corresponding Btrieve file.

Confirming the abort can leave the Btrieve file that is currently being rolled forward in an inconsistent state (that is, only partially updated). Select Yes to confirm you want to abort or No to cancel the abort and continue the roll forward process.

WBROLL–6.1–19: Log file name too long. Max = 120.

The MS Windows Roll Forward utility returns this message when the length of the log filename you specified is greater than 120 bytes. Specify an appropriate length for the log file.

WBROLL–6.1–20: This item or volume is already in the queue.

The MS Windows Roll Forward utility returns this message when the item or volume you entered in the queue has already been entered. Enter a different item or volume.

WBROLL–6.1–21: The listing length exceeds the buffer length.

The MS Windows Roll Forward utility returns this message when the specified listing length is invalid. In the Options dialog box, the Data to List option exceeds the value you entered for the Data Length option.

The Data to List option specifies the length of the data buffer that will be printed in the list file for each operation that is rolled forward. The Data Length option specifies the number of kilobytes allocated for the data buffer that the utility uses to process the logged entries. Make sure the value you enter for the Data to List option does not exceed the value you entered for the Data Length option.

WBROLL–6.1–22: The pathname is too long.

The MS Windows Roll Forward utility returns this message when the pathname you specified is greater than 240 bytes. Specify a shorter pathname.

WBROLL-6.10-23: An invalid path was specified.

The MS Windows Roll Forward utility returns this message when the path you specify is invalid. Specify a valid path.

WBROLL–6.1–24: Only one instance is allowed to run.

You tried to run the MS Windows Roll Forward utility, but it is already running.

Glossary

Accelerated

In pre-v6.*x* versions of Btrieve, a file open mode that provided improved response time when updating data files. Using Accelerated mode in pre-v6.*x* versions of Btrieve disabled Btrieve's data recovery capability.

In NetWare Btrieve v6.*x*, performance improvements for Normal open mode make Accelerated mode obsolete. Therefore, Accelerated mode is equivalent to Normal mode in NetWare Btrieve v6.*x*, except that opening a Btrieve file in Accelerated mode in NetWare Btrieve cancels the effect of flagging a file as transactional.

A file opened in Accelerated mode allows only one Btrieve client engine (and therefore only the clients associated with that engine) to access the file.

See also *Exclusive*, *file open mode*, *Normal*, *Read-Only*, and *Verify*.

all-segment null

A key attribute that instructs Btrieve to exclude a particular record from the index only if the value of all key segment of that record matches the specified null value.

See also *null key*.

any-segment null

A key attribute that instructs Btrieve to exclude a particular record from the index if the value of *any* key segment of that record matches the specified null value.

See also *null key*.

alternate collating sequence

A collating sequence that Btrieve uses to sort indexes containing a string type key. An alternate collating sequence allows Btrieve to sort the index differently than the standard ASCII collating sequence sorts them.

Also a key attribute that instructs Btrieve to sort the index using an alternate collating sequence when sorting by the key.

application

A program or set of programs, such as a spreadsheet or a payroll application, that performs a task or a group of related tasks. Also, a program written by or for a user to apply to the user's work.

application interface

A particular programming language interface (such as C or Pascal) that allows access to Btrieve files from an application.

ascending

A key attribute that instructs Btrieve to collate an index in ascending order. (For example, in an index that is sorted in ascending order for a string key, *a* would precede *b* and *b* would precede *c*.)

ASCII

An acronym for American Standard Code for Information Interchange. ASCII is a 7-bit information code that defines 128 standard characters, including control characters, letters, numbers, and symbols.

blank truncation

A method that Btrieve uses to conserve disk space by not storing the trailing blanks in the variable-length portions of records when the records are written to the file.

Btrieve

A key-indexed record manager for file handling. A function call from a standard programming language or from NetWare® SQL™ invokes Btrieve.

See also *client-based Btrieve*, *NetWare Btrieve*, *Record Manager* and *Requester*.

Btrieve file

A collection of records stored on a disk. A Btrieve file is sometimes referred to as a *physical file* or simply a *file*.

See also *data-only file* and *key-only file*.

Btrieve Requester

The Btrieve Requester program for the applicable DOS, OS/2, or MS Windows environment that resides at a workstation and provides communication between Btrieve and a workstation application making Btrieve calls.

buffer

An area of memory that is used temporarily to store data.

cache

The area of memory in which Btrieve stores images of physical disk pages. Btrieve uses the cache to reduce the number of physical disk I/O requests.

case insensitive

A key attribute that instructs Btrieve to sort an index so that values in uppercase letters are sorted in the same order as values in lowercase letters. (For example, *SMITH* is equal to *smith*.)

case sensitive

A key attribute that instructs Btrieve to sort an index so that values in uppercase letters are sorted before values in lowercase letters. (For example, *SMITH* would sort before *smith*.)

chunk

Any arbitrary portion of a record, specified by its offset and length. Btrieve 6.1*x* allows applications to update and retrieve variable-length records in chunks. An application that employs Btrieve's chunk feature can manipulate records greater than 64KB in length.

client

The meaning of the term client varies according to the context in which is used.

As used in *NetWare client-server environment*, client usually refers to a workstation that requests file or printing services from a NetWare file server.

However, client can refer to the Btrieve engine running on a particular workstation. In this situation, client refers to a single piece of software running on the workstation (as opposed to the workstation itself).

Client can also refer to each task currently running on a workstation. In a multitasking environment such as MS Windows, multiple clients can be active on the workstation at any given time. Therefore, when client is used in this manner, it usually identifies tasks that are instances of one or more Btrieve applications that are running simultaneously on a workstation.

Finally, client can refer to entities that are defined by the developer for use in a Btrieve application. The application then manipulates these entities by using the *BTRVID* or *BTRCALLID* functions. In this environment, multiple instances of an application can be running at the same time on a workstation, and each of those instances can have multiple Btrieve clients that it is manipulating.

client-based Btrieve

A version of Btrieve that runs on a platform other than NetWare (such as DOS, OS/2, or MS Windows).

All processing is performed on the workstation, and access to all files is through operating system calls. The operating system calls are either executed locally (for local files) or redirected to the server (for files on the server).

See also *Btrieve*, *NetWare Btrieve*, and *Record Manager*.

command file

A user-defined file that contains a sequence of commands.

commit

To save all the changes you have made to the database during a transaction.

See also *roll back* and *transaction*.

concurrency

The ability of multiple tasks to access the same data simultaneously while preserving data integrity.

See also *commit.*

concurrency controls

The methods Btrieve uses to resolve possible conflicts when two applications attempt to access the same data. Controls include passive control, record locking, and transaction control.

concurrent transaction

A type of transaction that allows other transactions to take place simultaneously in different parts of the designated file. A concurrent transaction implicitly locks the pages that it is modifying.

See also *exclusive transaction* and *implicit locks.*

configuration

The customization of a program such as Btrieve. You can customize Btrieve through the Setup utility.

configuration options

Specifications defined for Btrieve when it is loaded. Configuration options control the way in which Btrieve operates. You can use the Setup utility to change the configuration options.

connection number

The unique identifier number that NetWare assigns when a workstation attaches to a server.

continuous operation

A NetWare Btrieve feature that allows you to back up data files while they are open and in use. NetWare Btrieve opens the files in read-only mode to allow backup utilities to access the files' static images.

NetWare Btrieve stores changes to the original files in temporary files. When the backup is complete, NetWare Btrieve automatically updates the original files with the changes and deletes the temporary files.

cursor

See *position block*.

database

A set of one or more records or files that contain information on a related subject.

data buffer

A Btrieve function parameter that you use to transfer various information, depending on the operation being performed. A data buffer can contain all or part of a record, a file specification, and so on.

data buffer length

A Btrieve function parameter that you use to specify how much data is in the data buffer parameter.

data compression

A method for conserving disk space by compressing the space that repeating characters occupy.

data-only file

A Btrieve file in which no keys (and therefore no index pages) exist.

data page

See *page*.

deadlock

A condition that occurs when two tasks are retrying operations on files, pages, or records that the other one has already locked.

default

A preset option or value. For example, the default directory or disk drive is the one in which you are currently working.

descending

A key attribute that instructs Btrieve to collate an index in descending order. (For example, in an index that is sorted in descending order for a string key, *z* would precede *y* and *y* would precede *x*.)

description file

An ASCII file containing information that the Maintenance utility commands CREATE, INDEX, and SINDEX need to perform their operations.

directory

A disk structure that contains files. A directory may also contain subdirectories.

DOS

The DR DOS, MS-DOS, or PC-DOS operating system.

duplicatable

A key attribute specifying that multiple records in a file can have the same value in the key field.

See also *linked duplicatable* and *repeating duplicatable*.

dynamic link library (DLL)

A program library that contains related modules of compiled code. At runtime, the application reads the functions in the DLL. This process is called *dynamic linking*.

dynamic link routine

In OS/2 and MS Windows 3.*x*, a program that the operating system loads on demand (*dynamically*) and terminates automatically.

Exclusive

A file open mode. The user who opens a Btrieve file in Exclusive mode is the only one who can access that file.

See also *Accelerated*, *file open mode*, *Normal*, *Read-Only*, and *Verify*.

exclusive transaction

In Btrieve v5.*x*, a type of transaction that causes Btrieve to lock the entire file when a record is read, inserted, updated, or deleted. The Btrieve file remains locked until the task ends or aborts the transaction, or until NetWare times out the workstation's connection.

In Btrieve v6.*x*, a type of transaction that causes Btrieve to lock the entire file when a record is inserted, updated, or deleted. The Btrieve file remains locked until the task ends or aborts the transaction, or until NetWare times out the workstation's connection.

See also *concurrent transaction* and *implicit locks*.

explicit record locks

A type of concurrency control in which an application can explicitly lock one or more records by adding a lock bias value to the specified Btrieve operation. Explicit record locks lock the records so that the application can perform future update and delete operations.

See also *commit* and *implicit locks*.

extended operation

An operation that returns or inserts multiple records on one Btrieve call (for example, Get Next Extended or Insert Extended).

field

In Scalable SQL, a vertical collection of values in a table. All the values in a given field represent the same type of information. A field is also called a column.

In Btrieve, the term *field* has been used historically to refer to portions of a record that have been designated as segments of a key. Technically speaking however, Btrieve records contain no fields.

file

A collection of records stored on a disk. A file is sometimes also referred to as a *physical file* or a *Btrieve file*.

See also *data-only file* and *key-only file*.

file open mode

A method of opening a file that places restrictions on how that file can be accessed. The file open modes are Accelerated, Exclusive, Normal, Read-Only, and Verify.

See also *Accelerated*, *Exclusive*, *Normal*, *Read-Only,* and *Verify*.

filter

A restriction you can use when retrieving records with an extended operation.

fixed-length record

A record that contains no portions of variable length.

See also *variable-length record*.

free space threshold

A mechanism that Btrieve uses to determine whether to add data to an existing variable page or to create a new one. A higher free space threshold reduces fragmentation of variable-length records across several pages but uses more disk space. Btrieve also stores compressed records (even when fixed length) on variable pages.

implicit locks

The type of locking in which Btrieve automatically locks a record, page, or file according to the type of operation and transaction being executed by an application. Generally speaking, Btrieve locks a file during an exclusive transaction and a page or record during a concurrent transaction.

See also *commit,* *concurrent transaction*, *exclusive transaction*, *explicit record locks,* and *transaction*.

index

A structure in a Btrieve file that contains the key values for a specific key.

See also *key*.

index balancing

The process of searching for available space in sibling index pages when a given index page becomes full, and then rotating values from the full page into the pages that have space available.

integrity control

A method of ensuring that data in a file is complete and accurate. Btrieve uses concurrency controls and shadow paging to guarantee data file integrity.

See also *commit* and *shadow paging*.

key

A group of bytes (or multiple groups) that is characterized by offset and length (that is, by physical location in a record) and provides a means of direct access to a data value. In addition, a key provides a means of dynamic sorting of the records within a Btrieve file.

See also *index* and *segmented*.

key buffer

A Btrieve function parameter that usually contains the value of a key identified by the key number parameter.

key number

A Btrieve function parameter that usually identifies a specific key by which Btrieve is to retrieve a record from a Btrieve file. However, the information passed to or from Btrieve in the key number parameter depends on which operation is being performed.

For example, the key number parameter may indicate a value indicating the mode in which a Btrieve file is to be opened.

key segment

One of the groups of bytes that a segmented key includes.

See also *key* and *segmented*.

key-only file

A Btrieve file in which no data pages exist; that is, all records are stored on index pages.

key type

The internal format used by Btrieve when sorting records by the data in that key. Key types include `autoinc`, `bfloat`, `date`, `decimal`, `float`, `integer`, `logical`, `lstring`, `money`, `numeric`, `sign trailing separate`, `string`, `time`, `unsigned binary`, and `zstring`.

linked duplicatable

A key attribute that instructs Btrieve to store a duplicated key value on a data page as a pointer to the record containing the duplicate value. Btrieve stores the pointer at the end of the previous record containing the duplicated value (as opposed to storing the duplicated key value itself on an index page).

See also *repeating duplicatable*.

lock

A mechanism that prevents other tasks or clients from changing the data you are currently changing.

See also *commit, explicit record locks, implicit locks, no-wait lock*, and *wait lock*.

logging

A Btrieve utility that, when activated, records all the operations that change a specified Btrieve file. These changes are recorded in a log file. In the event of a system failure, Btrieve uses the log file to *roll forward* (recover) changes made to the Btrieve file between the time logging was initiated and the time of the system failure.

See also *roll forward*.

manual key

See *null key*.

modifiable

A key attribute that allows you to modify the key field during an update to a file. If a key is not modifiable, you cannot change the value in the key field.

NetWare Btrieve

A version of Btrieve that runs on the NetWare operating system (v2.*x* or later).

See also *Btrieve, client-based Btrieve,* and *Record Manager*.

Novell Directory Services™ (NDS™)

A system database that replaces the bindery used in previous versions of NetWare. NDS allows you to manage Directory objects, such as NetWare servers and volumes. NDS stores these objects in a hierarchical tree structure, thus providing a logical structure in which an object can reside apart from its physical location.

NetWare Loadable Module™ (NLM™)

A server program built into server memory with NetWare. You can load or unload an NLM while the server is running. The NLM becomes part of the operating system and can access NetWare directly.

Scalable SQL

A relational data access system, which resides at the server as an NLM, that allows users to run applications designed to manage shared data files.

NLM

See *NetWare Loadable Module™ (NLM™)*.

Normal

The default file open mode. In NetWare Btrieve, Normal mode allows shared read/write access to Btrieve files. With Normal open mode, Btrieve performs its standard integrity processing when it updates the data files.

See also *Accelerated*, *Exclusive*, *file open mode*, *Read-Only*, and *Verify*.

no-wait lock

A type of explicit record lock in which Btrieve returns control to your application if it cannot lock the record for any reason (for example, because another task has already locked that record). You request no-wait locks by specifying a lock bias value on a Btrieve operation code.

See also *explicit record locks* and *wait lock*.

null key

A key field that can be a user-defined character. Btrieve allows two types of null keys: any-segment (called a manual key in earlier versions of Btrieve) and all-segment (simply called a null key in earlier Btrieve versions).

An any-segment null key does not include a particular record in the index if the value of any key segment of that record matches the null value.

An all-segment null key only excludes a particular record from the index if the value of all key segment of that record matches the null value.

operation

A specific action that manipulates a Btrieve file (such as Delete, Create, or Get Equal). An operation is performed when an application calls one of the Btrieve functions.

owner name

A password that protects data files from unauthorized access by Btrieve applications. You can assign an owner name with the Btrieve Maintenance utility SETOWNER command.

page

A unit of a Btrieve file. A page is the smallest unit of storage that Btrieve moves between memory and disk. A page contains a multiple of 512 bytes (up to 4,096 bytes).

Btrieve uses the following types of pages:

- data page—contains fixed-length records (or the fixed-length portions of variable-length records)
- index page—contains key values and pointers to the associated records for those values (which reside on a data page)
- variable page—contains variable-length portions of records

See also *Btrieve file*, *fixed-length record*, *key*, and *variable-length record*.

parameter

An item of information that a program, utility, or API may need in order to perform a particular operation.

passive concurrency

A type of concurrency control in which your task does not perform any type of explicit locking. If another task modifies a record since you originally fetched it, you must fetch the record again before you can perform an update or delete operation.

See also *commit*.

pathname

The components that uniquely identify a file or directory.

For local workstation files, these components may include a drive letter, directory levels, and a filename.

For network files accessed from the workstation, these components might include:

```
[networkSpec] [pathSpec] [fileSpec]
```

- The `[networkSpec]` might contain the following subcomponents:
  - A network drive letter
  - A server name and volume name
  - A volume name (current server is assumed)
  - Nothing (current server and volume is assumed)
- The `[pathSpec]` might contain the following subcomponents:
  - A complete directory path
  - A relative directory path
  - Nothing
- The `[fileSpec]` might contain a filename and possibly an extension.

position block

An internal Btrieve structure that keeps track of an application's position in a file when reading or updating records. Also one of the parameters required when calling a Btrieve function.

preallocation

A method for allocating a specific amount of disk space for a Btrieve file when you create that file. This space is reserved for future expansion of the file when Btrieve needs it.

pre-imaging

Storing the image of a file page before updating a record on the page. Btrieve 5.*x* and earlier use pre-imaging to provide recovery capabilities in case a file is damaged or a system failure occurs during an update to the file.

Read-Only

A file open mode that does not allow you to insert, update, or delete records.

See also *Accelerated*, *Exclusive*, *file open mode*, *Normal*, and *Verify*.

record

A set of logically associated data items in a Btrieve file. Generally, a record is the unit transferred between an application and Btrieve in a single operation.

Btrieve sees each record as a collection of bytes and performs no translation or type verification of the data in a record. The application interfacing with Btrieve must handle all information about the format and type of the data in records.

record locking

A type of concurrency control that enables an application to lock the record it is accessing within a file. Other users can read the record, but no other user can lock, update, or delete the record until the application that holds the lock releases it.

Record Manager

The part of Btrieve that maintains the records in a data file. For example, the Record Manager opens the data file and retrieves, modifies, inserts, and deletes records.

The server-based Btrieve Record Manager is a program that resides at the server and handles data I/O with the file system. The client-based Btrieve Record Manager resides at the workstation and handles data I/O with the file system through operating system calls.

See also *client-based Btrieve*, *NetWare Btrieve*, and *Requester*.

referential integrity (RI)

In Scalable SQL, the assurance that when a field in one table references a field in another table, changes to these fields are synchronized.

repeating duplicatable

A key attribute that instructs Btrieve to store a duplicated key value on an index page.

See also *linked duplicatable*.

Requester

The portion of NetWare Btrieve that resides at a workstation and passes Btrieve requests from an application to the Btrieve Record Manager on the NetWare file server.

requester

A program that resides at a workstation and passes requests from an application to a server-based application.

requester interface

A program that resides on a workstation and passes requests from a client application to a requester.

RI

See *referential integrity (RI)*.

roll back

To abort a transaction and undo all the changes made to a Btrieve file during the transaction, thus restoring the file to the state it was in before the transaction began.

See also *commit* and *transaction*.

roll forward

Recovering changes made to a Btrieve file between the time logging is initiated and a system failure.

See also *logging*.

roll in

Writing to a Btrieve original file all the changes made to the corresponding temporary file during the continuous operation backup period. When the backup is complete, Btrieve automatically updates the original file with the changes and deletes the temporary file.

See also *continuous operation.*

segmented

A key attribute that allows a key to consist of more than one part of the record. A segment can be any set of contiguous bytes in a record. Segments within a record can vary in type and/or length, and need not be contiguous to other segments of the key.

Sequenced Packet Exchange (SPX)

A Novell communication protocol that monitors network transmission to ensure successful delivery. SPX runs on top of Novell's Internetwork Packet Exchange (IPX).

shadow paging

A technique that Btrieve 6.*x* uses to ensure data integrity. When a user needs to change a page, Btrieve makes the change to a physical shadow page, which is a virtual copy of the original page. When the change is committed, Btrieve designates the shadow page as the current page, and the original page becomes available for reuse.

sign trailing separate (STS)

A COBOL data type that is basically a numeric data type, represented as an ASCII string. `STS` is right justified and padded with leading zeros, and it has the sign byte at the end.

SPX

See *Sequenced Packet Exchange (SPX)*.

string

A series of characters (as opposed to a number), or a category of data types used to store strings. This category includes `string`, `lstring`, and `zstring`.

supervisor

The person responsible for the administration and maintenance of a network, a database, or both. A supervisor has access rights to all volumes, directories, and files.

supplemental index

In Btrieve v5.*x*, an index added to the file after it was created. You could delete a supplemental index, but you could not delete an index that was created when the file was created. In Btrieve v6.*x*, supplemental indexes are no longer used because you can delete all types of indexes.

table

In Scalable SQL, a collection of data formatted in rows, where each row consists of field values. Vertical collections of field values in the table form fields, and each field contains one type of information, such as salary or revenue. Each row contains the same number of items and type of information as every other row in the table.

task

An instance of an application.

See also *application* and *client*.

thread

A separate stream of execution within a program.

transaction

A set of related operations that constitutes a logical unit of work. The application performing the transaction requires that either all or none of these operations be performed.

Transaction Tracking System (TTS)

A NetWare fault tolerance system that protects files (including Btrieve files) from corruption by backing out incomplete transactions that result from a failure in a network component.

Versions of NetWare Btrieve prior to 6.0 use TTS to protect Btrieve files on a NetWare server. TTS must be active at the server, and the Btrieve files must be flagged as transactional.

NetWare Btrieve v6.*x* uses TTS somewhat differently. Applications can flag a file as transactional when they want to signal NetWare Btrieve that the file's integrity *must* be guaranteed.

However, Btrieve does not actually use TTS to ensure integrity because TTS is not as fast as other means when used in conjunction with concurrent transactions. Instead, when Btrieve v6.1 notices the TTS flag, it forces pages to be written chronologically (in order to ensure that Btrieve recovery mechanisms work properly).

Client-based versions of Btrieve disregard the TTS flag.

TTS

See *Transaction Tracking System (TTS)*.

user

Someone who is authorized to log in to a network and/or database when security is installed and who has access rights to specific filenames, directories, and files.

variable-length record

A record that contains a variable-length portion and a fixed-length portion. While the fixed-length portions must be the same size in all the records in a given file, the variable-length portions may vary in size. This means the overall lengths of variable-length records may vary.

See also *fixed-length record*.

Variable-tail Allocation Table (VAT)

An array of pointers to locations within the variable-length portion of a record.

variable page

See *page*.

Verify

In DOS-based Btrieve v5.x, a file open mode that causes the operating system to verify all write operations to a file. Opening a file in Verify mode in Btrieve v6.x has no effect other than to open that file in Normal mode.

See also *Accelerated*, *Exclusive*, *file open mode*, *Normal*, and *Read-Only*.

wait lock

A type of explicit record lock in which Btrieve does not return control to your application until it has obtained the lock on the record you requested. You request wait locks by specifying a lock bias value on a Btrieve operation code.

See also *explicit record locks* and *no-wait lock*.

Index

User Comments

We want to hear your comments and suggestions about this manual. Please send them to the following address:

Novell, Inc.
Documentation Development
PRV-C231
122 East 1700 South
Provo, UT 84606
U.S.A

e-mail: commentdoc@novell.com

Fax: (801) 861-3002

NetWare 4.11
Btrieve Installation and Operation
Part #100-003597-001 A
September 1996

For technical support issues, contact your local dealer.

Your name and title: ____

Company: ____

Address: ____

____

____

Phone number: ______________________________ Fax: ____________________

I use this manual as ❑ an overview ❑ a tutorial ❑ a reference ❑ a guide ❑ ____________

	Excellent	Good	Fair	Poor
Completeness	❑	❑	❑	❑
Readability (style)	❑	❑	❑	❑
Organization/Format	❑	❑	❑	❑
Accuracy	❑	❑	❑	❑
Examples	❑	❑	❑	❑
Illustrations	❑	❑	❑	❑
Usefulness	❑	❑	❑	❑

Please explain any of your ratings: ____

____

____

____

____

In what ways can this manual be improved? ____

____

____

____

____

You may photocopy this comment page as needed so that others can also send in comments.