

File Indexing Enhancements in NetWorker 6

File Indexing Enhancements in NetWorker 6

Improved Performance, Scalability and Robustness

File Indexing Enhancements in Networker 6

Improved Performance, Scalability and Robustness

Table of Contents

1 Executive Summary 3

2 Growth in Data Storage Tests the Limits of Backup Software 3

3 Index Enhancements to Improve Performance, Scalability, and Robustness 4

3.1 Performance Improvements 6

3.1.1 Decreased Index Size 6

3.1.2 XDR Encoding 7

3.1.3 Backup and Recovery of Client File Indexes 7

3.2 Enhancements for File Index Management 7

3.3 Disaster Recovery 8

3.4 New Features 9

3.4.1 Index Roll-in 9

3.4.2 Finer Grained Control of Browse and Retention Policies 9

3.4.3 Cross-platform Browsing 9

3.4.4 NDMP Support 9

3.5 Upgrading to NetWorker 6.0 9

4 NetWorker 6.0 Performance 10

4.1 Customer Experience 10

5 A Foundation for the Future 11

Appendix—Detailed Performance Comparisons 12

1 Executive Summary

The explosive growth in data storage has resulted in extreme stress on backup software. Recognizing that the file indexing methodology used in NetWorker was reaching the limits of the original design, Legato completely re-architected the file index for NetWorker 6.0. The new design substantially improves performance in many critical areas. File index backup now occurs at device speed. Browsing through saved files is now 300% faster. The file index has been re-designed to use 20%-40% less space than before, and common management tasks can be completed more than five times faster. NetWorker 6.0 also provides important new features like index roll-in, NDMP, cross-platform data browsing, and more flexible policies for stored data.Growth in Data Storage Tests the Limits of Backup Software

2 Growth in Data Storage Tests the Limits of Backup Software

The past few years have witnessed remarkable growth in online data storage, fueled by increased storage density, reduced storage prices, and new applications. Many large enterprises are doubling online storage capacity every year. The rise of the Internet has resulted in entirely new types of businesses with voracious appetites for storage. These businesses—providing Internet access, applications, web hosting, infrastructure and a variety of other services— are projected to grow at rates in excess of 150% over the next several years. For many, storage needs will grow at an even faster rate. Effectively protecting important data with regular backups is a strategic advantage in an industry subject to consolidation and outright failure.

Protecting ever-expanding pools of online storage with regular backups is becoming increasingly challenging. Backup tools designed to handle tens or hundreds of gigabytes of storage are being overtaxed by online storage reaching into the terabytes. The file indexes that backup tools use to record information on each backed up file have grown to occupy many gigabytes of storage, taxing internal algorithms, reducing performance, and limiting scalability.

Legato Systems has designed NetWorker 6.0 to work robustly and efficiently even when total storage reaches many terabytes. File indexing has been completely re-architected to improve performance, parallelism and robustness. Key performance improvements include:

- Reduction in the size of file indexes, typically by 20% to 40% compared with previous versions
- Better scaling for concurrent index operations
- More rapid backup and recovery of file indexes
- Faster browsing
- Index performance scales linearly with the number of savesets

As file indexes grew larger, regular index management tasks became increasingly cumbersome. NetWorker 6.0 provides substantial improvements for index management including:

- Better scaling/support for very large client file indexes
- Nearly instantaneous crosschecks
- Simplified relocation of client indexes to new file systems
- Elimination of file index corruption

When disaster occurs, getting a backup server up and running so that other systems can be recovered is the first priority. Enhancements in NetWorker 6.0 allow more rapid creation of bootstraps, and bootstraps are now completely self-contained. (A bootstrap contains the site specific information that a server needs to get up and running after a disaster.) File recovery or regular backups can commence in a matter of minutes.

Many new features were added to NetWorker 6.0, including:

- Index roll in
- Finer grained control over browse and retention policies (per saveset)
- Support for cross-platform browsing of backups
- NDMP support

Upgrading to NetWorker 6.0 from previous NetWorker releases is fast and simple. File indexes are converted automatically and the performance of the conversion can be configured for the underlying hardware. Backward compatibility ensures that an upgraded server will continue to work correctly with clients running previous releases.

This paper discusses the enhancements made to the indexing system that result in these dramatic improvements. A detailed performance comparison with NetWorker 5.5 is included.

3 Index Enhancements to Improve Performance, Scalability, and Robustness

Legato NetWorker utilizes a client/server architecture. A central server—running NetWorker server software—controls all backup operations. Data to be backed up from clients—running NetWorker client software—is sent across the network to storage devices attached to the server or to a Storage Node—a server-managed system that hosts storage devices. Client data is backed up in increments called *savesets*. A saveset typically consists of all the data to be backed up from a single file system or logical volume.

All information about files being backed up—including filename, time of back up, size, permissions, and ownership—is directed to the NetWorker server and saved in a *client file index*. In addition to the client file index, the server also maintains the *media database*. The media database keeps track of the savesets on each storage volume and contains browse and retention policies.

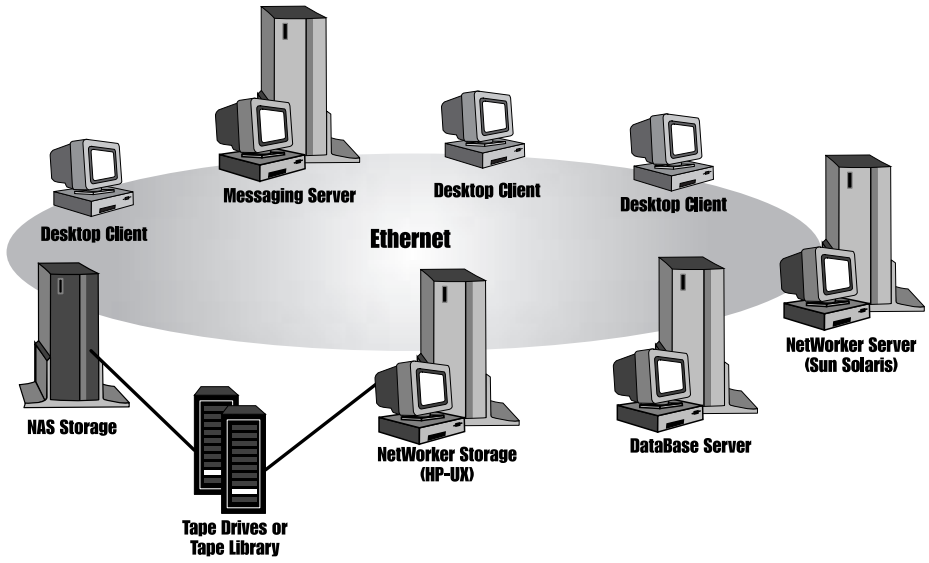


Figure 1. **NetWorker system architecture**
Architecture shows NetWorker server, Storage Nodes. Clients, NAS device and shared tape library

Prior to NetWorker 6.0, the client file index consisted of a large, monolithic database for each backup client. This database could become quite large and cumbersome to manage for clients with a large amount of storage. For instance, as savesets timed out and were discarded, records had to be removed. The gaps created by large deletions had to be recovered through a special process called index compression that was very time consuming. (NetWorker 6.0 eliminates the need for compression.)

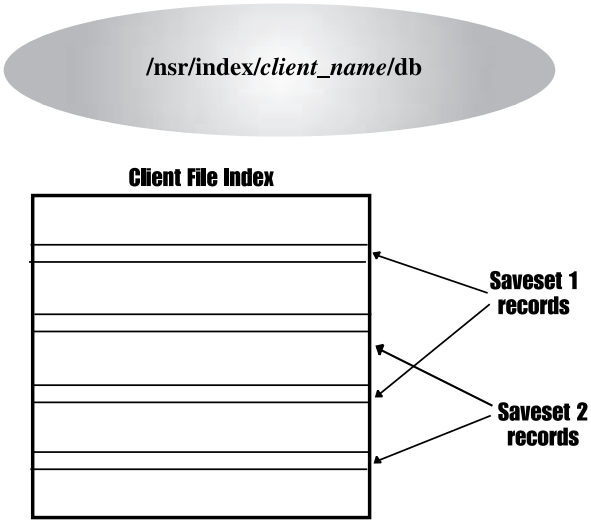


Figure 2. **The Client File Index prior to NetWorker 6.0.**
Each time a saveset is backed up the individual file records are inserted into a single large database. Since multiple savesets are typically backed up in parallel, the file records are interleaved. In addition, the database contains records from multiple backups of a single client's savesets.

The operations performed on the index data are highly specialized. In NetWorker 6.0, the client file index database has been completely re-designed and optimized based on the relative frequency of index operations. From most to least frequent, these are:

1. Add a set of records comprising a saveset
2. Backup recently added savesets
3. Lookup individual records
4. Purge recently timed-out savesets
5. Check consistency
6. Recover a set of savesets/disaster recovery

Optimizing the organization of the client file indexes dramatically streamlines these operations, especially those performed most frequently. Instead of maintaining a single large database that must grow and shrink as records are added and removed, a separate index file is now created in the database directory each time a saveset is backed up. Adding a set of records for a saveset, backing up recently added saveset records and purging timed-out savesets thus become simple operations on individual files. Two key files are also generated per index file. These are organized by filename and fileid and are used to accelerate access to individual records.

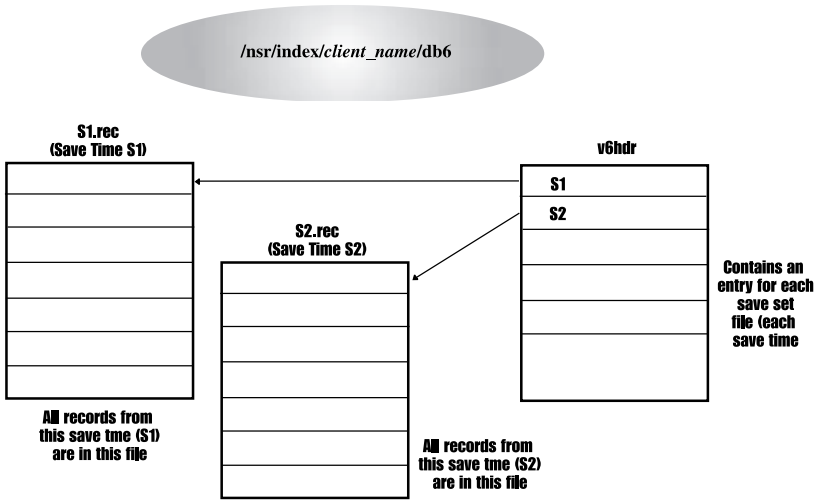


Figure 3**. Client File Index organization in NetWorker 6.0.**
Each time a saveset is backed up a separate index file is created. The file *v6hdr* keeps track of all the index files stored in the directory. Two key files are also created to accelerate access to file records (not shown).

Client file index record files are only written once, at the time the saveset is created. After creation, they are read-only. This write-once read-many management approach simplifies locking, minimizes resource contention, and increases efficiency and availability. This makes the software much more robust.

3.1 Performance Improvements

The optimization of the client file index results in significant performance benefits. Busy clients frequently have multiple operations in progress that require reading or writing the client file index. In previous releases, a bottleneck occurred at the client file index that could delay backup progress. In NetWorker 6.0, records for each saveset are written to a separate file, allowing for much greater concurrency. Index performance scales linearly with the number of savesets. Browsing a saveset is dramatically accelerated since the new index mechanism is optimized for the most frequent operations on the data. The old mechanism was inefficient by comparison.

3.1.1 Decreased Index Size

Several additional optimizations were made to the design of the client file index. The most significant change results in a substantial decrease in size. In previous versions, the full pathname for each file was included in its index record. In NetWorker 6.0, each save record now contains a pointer to the appropriate pathname within a set of pathnames stored with the saveset. (This list is referred to as the Directory Prefix Table or DPT). Effectively, this is a table of all directories that contained one or more files that were backed up in that saveset. (Note that this is not necessarily a complete set of all directories in the original volume, since an incremental backup might skip many directories.)

This change typically results in a 20% to 40% decrease in the size of the client file index depending on the number of files stored in each directory and the length of the pathnames. For very long pathnames and directories with large numbers of files, the space savings could be even greater

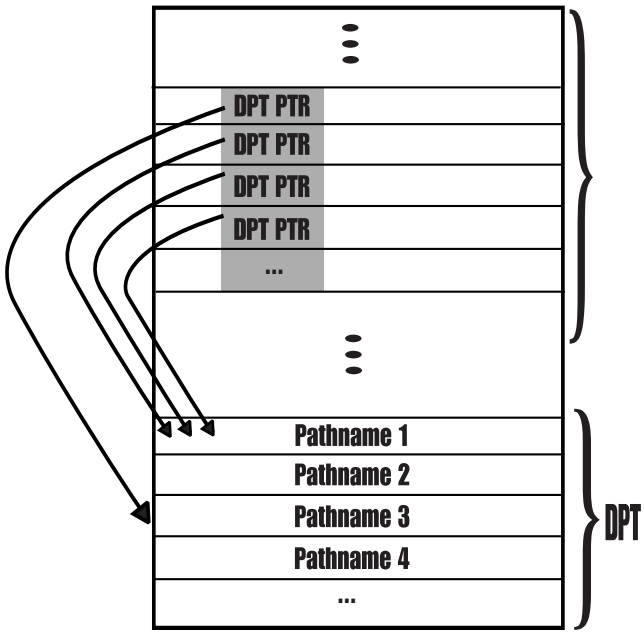


Figure 4. **The Directory Prefix Table (DPT).**
Each file record in the index file has a pointer to the appropriate pathname in the DPT. By eliminating redundancy, this reduces the size of the file by 20% to 40% or more depending on the length of the pathnames and the number of files per directory.

3.1.2 XDR Encoding

In NetWorker 6.0, the client file indexes are XDR-encoded as they are written. XDR (eXternal Data Representation) encoding allows file save records to be read on machines of any architecture, which can be critical for disaster recovery. Formerly, file save records were only XDR-encoded when they were backed up to storage media. XDR-encoding the client file indexes streamlines backup and recovery of these files and allows them to be transferred directly between machines of differing architectures when necessary.

3.1.3 Backup and Recovery of Client File Indexes

Backup and recovery of the client file indexes themselves was formerly a complicated and slow operation since it required scanning a large database, extracting or adding records, and encoding or decoding with XDR. In 6.0 the file containing the desired saveset index is backed up or recovered as with any other file. Since the files contain records from a single saveset and are only written once, it is trivial to identify the files that must be backed up when saving the client file index. Since the file is already XDR-encoded no additional steps are needed. Thus, all past and future improvements made in file backup performance now directly improve client file index backup performance as well.

3.2 Enhancements for File Index Management

The re-design of the client file index mechanism also provided the opportunity to make some major improvements to index management. Based on the discussion above, it should be clear that NetWorker 6.0 provides better support for clients with very large data sets. Managing the client file indexes for such clients is much easier with the data distributed across multiple files. Many operations that were time consuming and cumbersome now occur almost instantaneously.

Crosschecking is a good example of improved efficiency. The media database stores the data retention policies. When a saveset expires and is no longer browsable, the client file index must be reconciled with the media database and all records associated with that saveset must be deleted. Prior to 6.0, this required a lengthy scan of the monolithic client file index database to identify and remove all records associated with that saveset. The removal of records from the database left holes (unused space) that could only be recovered by compression. In NetWorker 6.0, crosschecking simply requires identifying the client file index associated with the expired saveset and deleting the entire file. The new process is nearly instantaneous, occurs automatically, and doesn't require a special compression process to recover unused space. Previously, the index entries weren't removed until a crosscheck was run (typically once a day), so there was a period of time when the media database and client file index were inconsistent.

In previous releases, client file indexes also were subject to corruption. Certain processes could terminate abnormally, leaving the index in an inconsistent state. Consistency checking was complicated and performance degraded exponentially with the number of records. Because the index files are now written only one time and are subsequently read-only, index corruption can only occur if the underlying file system is somehow corrupted. There are never multiple processes contending for write access to the file. As the backup of a saveset completes, a checksum is calculated for the entire index file corresponding to that saveset. Consistency can be verified simply by re-calculating the checksum and comparing it to the stored value.

Relocating client file indexes or changing a client's name was formerly problematic because the file indexes themselves were backed up using the full path name. Records indicating that a client file index was backed up were stored within the server's client file index. In NetWorker 6.0, only the media database stores information on the back up of client file indexes. In addition, the saveset name for all client file indexes has been normalized to:

index:*clientid*

Where **index** designates that the saveset represents an index file and ***clientid*** is a unique identifier for the client. To relocate an index it is only necessary to move the files and update the client resource attribute ***index path*** **to reflect the new location**.

One side effect of this change is that it is no longer possible to browse the server's file index to recover a client file index using the standard browser. Index recovery must, instead, be performed using the **nsrck** utility. The benefits of this change in terms of flexibility far outweigh this minor inconvenience.

3.3 Disaster Recovery

When a disaster occurs, the first priority is to restore the NetWorker server so that client recovery can begin. This is done by recovering a *bootstrap* containing all the site-specific information the server needs to function. In NetWorker 6.0, all data required to perform backups is stored in the media database and resource database. Backing up the server’s index of saved client file indexes during bootstrap generation is unnecessary. Disaster recovery is simplified because the bootstrap saveset contains only the media database and resource files. Only this saveset need be recovered before the server is again operational. Regular backups and recovery by saveset can be resumed in a matter of minutes once the bootstrap is recovered. File-level recovery for a client can begin once that client's file index is restored.

3.4 New Features

The re-design undertaken for NetWorker 6.0 also provided the opportunity to implement new and often requested features.

3.4.1 Index Roll-in

Index roll-in allows a client file index to be restored to any designated point in the past. In previous releases, restoring a saved version of the client file index was destructive. Since the file-index was treated as a monolithic file, it was overwritten and all changes that occurred after the time of the restored index were lost. With the new design, client file indexes from old savesets can be restored and happily co-exist with newer data.

3.4.2 Finer Grained Control of Browse and Retention Policies

NetWorker uses browse and retention policies to manage the life cycle of data and to control the size of the client file index automatically. The browse policy determines how long files are maintained in the client file index on the server. The retention policy determines how long savesets are maintained in the server's media database.

In earlier NetWorker releases, all savesets for a given client had to have the same browse and retention policy. With NetWorker 6.0, different browse and retention policies can be specified for each saveset. You can also change the browse and retention policies of savesets at any time, without affecting the policies in force on other savesets. More flexible browse and retention policies allow customers to create daily, monthly, and yearly backups more easily than was previously possible.

A new field, insertion time, has been added to the records in the media database. This field records the time the saveset was inserted. It is different from the savetime in cases where the saveset timed out and was later read back in. All browse and retention policies are now based on the insertion time. This allows browse and retention policies to be set for scanned in savesets so they expire and are removed automatically. This was not previously possible.

3.4.3 Cross-platform Browsing

Cross-platform browsing also was added in NetWorker 6.0. Used in conjunction with directed recovery, this feature allows Unix systems to browse NT backups and recover those backups to the same or different NT system as long as NetWorker is installed. Likewise, NT systems can browse Unix backups and recover them to the same or different Unix client.

3.4.4 NDMP Support

The Network Data Management Protocol (NDMP) is a standard protocol for controlling data and tape operations on network-attached devices. The architectural changes made in NetWorker 6.0 included expanding several fields in the file index and media database to accommodate NDMP data types. (A complete discussion of the changes made to support NDMP is beyond the scope of this document.) Using NDMP, NetWorker 6.0 can backup, restore and/or control NDMP-compliant network devices running version 2 or version 3 of NDMP.

3.5 Upgrading to NetWorker 6.0

Upgrading directly to NetWorker 6.0 is supported for releases 4.2 and later. NetWorker is able to upgrade the file index format and has the ability to recover old backups of the file index. Conversion of client file indexes and media database is initiated automatically when upgrade begins. There is no provision to convert back to the old format other than recovering from an old backup. New backups can begin as soon

as the media database is converted (typically a half hour or less). On multiprocessor systems, the number of processes dedicated to file index conversion can be increased, decreasing the time needed for conversion to complete.

Upgraded servers can interoperate with clients running older versions of software. Once a tape volume has been written by a server or Storage Node running NetWorker 6.0, it cannot be read by servers or Storage Nodes running older versions of software.

4 NetWorker 6.0 Performance

The changes described in the preceding section have resulted in significantly improved performance in all areas measured. The performance of NetWorker 6.0 was directly compared to that of NetWorker 5.5 in laboratory testing. All performance measurements were made using a Sun Ultra2 with a single 296MHz Ultra SPARCII processor, 640MB of memory, and a single 10,000 RPM disk drive. The system was configured to run either NetWorker 5.5 or 6.0. A file system was created containing 14 million zero-length files. All backup data was directed to file devices rather than tape to eliminate the overhead of tape head positioning. These conditions were imposed to ensure that stress on the file index mechanism was maximized.

The following table summarizes the results:

Test	Performance Improvement
Browsing	300% faster
File Index Recovery	275% faster
Index Management Operations (crosscheck, compression, correction)	A minimum of 575% faster
Index Record Insertion	37% faster
Full Index Backup	Device speed versus maximum of 2.5MB/sec
Incremental Index Backup	Device speed and independent of index size (5.5 performance decreases as index size increases)
File Index Size	20% to 40% smaller

Table 1: Performance comparison of NetWorker 6.0 to Networker 5.5

As the above table clearly indicates, NetWorker 6.0 demonstrated dramatically improved performance in all areas tested. A complete summary of performance comparisons is provided in the Appendix at the end of this document.

4.1 Customer Experience

Testing and use of NetWorker 6.0 at various customer sites confirms the results of Legato's laboratory testing. The noticeable decrease in file index size has frequently been mentioned as a substantial benefit. The following table summarizes the results of converting client file indexes at one customer site. The NetWorker server was running Solaris 2.7 with four 300MHz processors, 1GB of memory, two internal 18GB drives and twelve SAN-attached 7.2 GB drives.

Client File Index	Initial Index Size (KB)	Final Index Size (KB)	Percent Reduction	Conversion Rate
1	112360	68312	39%	1.8MB/sec
2	197992	156968	21%	1.8MB/sec
3	66056	46960	29%	1.8MB/sec
4	3291864	2295760	32%	1.5MB/sec

Table 2: Reduction in Index Size as measured at a Customer Site.

These results are completely consistent with the 20% to 40% reduction in size measured in laboratory tests. The largest index—with approximately 3GB of client file index data—took 36 minutes to complete the conversion process, so even very large indexes are converted at a rate in excess of 6GB per hour.

Upgrade performance for NetWorker 6.0 is substantially improved over previous releases. At one customer site, a 444MHz, dual-processor Solaris system with 5,000 volumes and 510,000 savesets was upgraded in 19 minutes. Another site with a 168MHz, eight-processor NetWorker server also took only 19 minutes to upgrade 3,000 volumes and 450,000 savesets. This means that in only 19 minutes, these customers were able to upgrade and resume normal operations.

5 A Foundation for the Future

The enhanced indexing in NetWorker 6.0 eliminates the bottlenecks that became apparent with the explosive growth of customer data in recent years. Performance and scalability have dramatically improved and—because many common operations are now simple to perform—the design is much more robust. Large enterprises will benefit from substantial operational improvements and greater flexibility in managing servers and clients. NetWorker will scale to meet the needs of the enterprise for many years to come. Extensive platform and database support ensures that NetWorker can accommodate a wide array of backup needs. The addition of NDMP further extends this capability, allowing NetWorker to backup network attached storage devices or other systems compliant with NDMP version 2 or version 3.
For companies on the cutting edge of the Internet, NetWorker's scalability and performance can be a strategic advantage. Simple to install and easy to use, NetWorker can grow quickly and flexibly to accommodate explosive growth.

Appendix—Detailed Performance Comparisons

All performance measurements were made using a Sun Ultra2 with a single 296MHz Ultra SPARCII processor, 640MB of memory, and a single 10,000 RPM disk drive. The system was configured to run either NetWorker 5.5 or 6.0. A file system was created containing 14 million zero-length files. All backup data was directed to file devices rather than tape to eliminate the overhead of tape head positioning.

Operation	5.5.1	6.0
Full savegroup		
(Save 14 million empty files)	03:44:22	02:44:21
Index Save		
Time	00:33:26	00:19:05
Records	27,891,799	41,852,453
Size	5018 MB	4828 MB
Speed	2.5 MB/sec	Device Speed
Bootstrap Save		
Time	00:07:17	00:00:02
Size	42 KB	41 KB
Speed	98 bytes/second	20 KB/second
Incremental savegroup		
(Save /etc)		
Index Save		
Time	00:07:17	00:00:01
Size	7 KB	5 KB
Speed	16 bytes/second	Device speed
Bootstrap Save		
Time	00:07:17	00:00:01
Size	42 KB	35 KB
Speed	98 bytes/second	Device Speed
Browse performance		
(“add” 1,113,223 files)		
Time	00:50:42	00:14:42
Disaster Recovery		
Data Recovery		
Media Database	Unknown	00:00:27.8
File Index	Unknown	00:17:09
Rebuild server file index keys		
Time	5:09:51	00:54:03
Records	27,891,799	27,891,805
Size	6548 MB	3978 MB
Total Time	6:15:52	1:11:39
Total Time to being able to run backups:	6:15:52	similar to media db upgrade
nsrck –X (remove nothing)		
Time	00:07:23	00:00:00.1
Records	27,891,799	27,891,805
Size	6548 MB	3978 MB
nsrck –X (remove ~26,000,000 records)		
Time	13:47:55.1	00:00:03.4
Records deleted	26,748,271	27,891,294
Rebuild keys	nsrck -F	nsrck -L6
Time	5:09:51	00:54:03
nsrck –C (compress)		

Time	more than nsrck –F	N/A
nsrck –L5 (rigorous consistency check) Time	N/A	00:09:56
nsrck –L1 through -L4 Time	N/A	less than 1 second



Legato Systems, Inc.
2350 West El Camino Real
Mountain View, CA 94040
USA
Tel (888) 853.4286
Tel (650) 210.7000 • Fax (650) 210.7032
www.legato.com
For a complete listing of Legato Systems offices worldwide, please visit
http://www.legato.com/About/index.html