

Advantage Database Server

User Manual

for Novell NetWare® and Windows NT®

January 2004

NDCHEALTH™

5222 E. Baseline Rd., Suite 101
Gilbert, AZ 85234

Support	(800) 333-4747
Sales	(800) 334-4006
Corporate	(480) 892-5120
Fax	(480) 892-4804
Website	www.medisoft.com

Contents

PREFACE	I
License Agreement.....	i
Disclaimer	i
Permitted Uses	i
Prohibited Uses	ii
NDCHealth's Rights	ii
Trademarks	ii
Limited Warranty.....	iii
Termination of License.....	iii
Miscellaneous	iv
Technical Support	iv
■ Independent NDCMedisoft Value-Added Reseller.....	iv
■ NDCMedisoft Technical Support	iv
CHAPTER 1	1
INTRODUCTION.....	1
The Advantage Database Server	1
Support for NetWare and Windows NT/2000 Operating Systems	2
The Advantage Product Family	3
Advantage Client Kits.....	3
CHAPTER 2.....	5
INSTALLING THE ADVANTAGE DATABASE SERVER.....	5
System Requirements	5
■ Networks.....	5
■ Network Protocols (on the Server)	5

■ File Server Processors	5
■ Client Operating Systems.....	5
■ Client Development Platforms.....	6
■ Xbase Table Formats Supported.....	6
■ Xbase Index File Formats Supported.....	6
■ Xbase Memo File Formats Supported	6
■ Proprietary Advantage Table Format Supported.....	6
■ Proprietary Advantage Index File Format Supported.....	7
■ Proprietary Advantage Memo File Format Supported	7
■ Licensing	7
Installing and Loading the Advantage NLM for NetWare	7
Installation.....	7
Loading the Advantage NLM.....	8
Unloading the Advantage NLM.....	8
Installing the Advantage Service for Windows NT/2000	9
Understanding Windows NT/2000 Services	9
Advantage Database Server Service	10
Starting and Stopping the Advantage Service.....	10
To start and stop the Advantage Database Server Service for Windows	
NT 4.0:	10
To start and stop the Advantage Database Server Service for Windows	
2000:	11
Using Windows NT/2000 Event Logging.....	11
Event Logging and the Advantage Database Server Service	11
CHAPTER 3	13
ADVANTAGE FUNCTIONALITY.....	13
Client/Server Technology	13
Multi-User Performance	14
Optimized Data Access	14
Intelligent Lock Management	15
Database Stability	16
CHAPTER 4	19
ADVANTAGE DATABASE SERVER FEATURES	19
Database Security.....	19
Check Rights	19
Ignore Rights	20
Configured Semaphore Connection File Directory Often Necessary with	
the Advantage Database Server.....	22

Advantage Locking Modes.....	23
Advantage Proprietary Locking.....	23
Advantage Compatibility Locking with Xbase Files.....	24
Advantage Compatibility Locking with Advantage ADT Files	25
Single Application Scalability.....	25
Reduced File Handle Usage.....	26
Advantage Optimized Filters	27
Differences Between AOFs and Traditional Record Filters	29
AOF Optimization	30
AOF Relational Operators.....	32
AOF Performance Tips	33
Read-Ahead Record Caching.....	34

Preface

License Agreement

NDCHealth Corporation (“NDCHealth”) grants to the original purchaser (“Purchaser”) a nontransferable, nonexclusive license to use the accompanying software (the “Software”) in machine-readable, object code form only, and the associated documentation for the private use of Purchaser as set forth in the Permitted Uses and Prohibited Uses sections below. Acceptance and use of the Software acknowledges Purchaser's agreement to abide by the terms of this License Agreement.

Copyright © 1995-2004 Extended Systems Incorporated. All rights reserved.

Disclaimer

NDCHealth makes no representations or warranties with respect to the contents of the Software and specifically disclaims any implied warranties and warranties of merchantability or fitness for any particular purpose. NDCHealth reserves the right to make changes or alterations in the Software without the obligation to notify anyone of the changes or alterations.

Permitted Uses

Purchaser may (1) copy the Software for backup purposes to support Purchaser's use of the Software; (2) use the Software on a single computer; and (3) if integrated with NDCHealth networked software licensed by Purchaser

from NDCHealth, Purchaser may install this Software on each computer within one network at a single location.

Prohibited Uses

Purchaser may not (1) distribute, rent, sublicense, or otherwise make available to others the Software, documentation, or any copies thereof; (2) modify, translate, adapt, disassemble, or create derivative works from the Software or documentation; or (3) use the Software or permit it to be used on more than one computer at any one time, except as expressly authorized in the Permitted Uses Section above. The Software, documentation, and all manuals accompanying the Software are copyrighted and all rights are reserved by NDCHealth. No copyrighted material may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable device without prior written consent from NDCHealth.

NDCHealth's Rights

Purchaser acknowledges and agrees that the Software and associated documentation are proprietary products of NDCHealth and certain third parties ("Third-Party Licensors") protected under U.S. and international copyright law. Purchaser further acknowledges and agrees that all right, title, and interest in and to the Software and any copies thereof, including associated intellectual property rights, are and shall remain with NDCHealth and Third-Party Licensors. The License Agreement does not convey to Purchaser any interest in or to the Software, but only a limited right of use revocable in accordance with the terms hereof, and Purchaser agrees to transfer to NDCHealth any property interests or rights in the Software Purchaser may have or later obtain by operation of law or otherwise which are inconsistent with the rights granted to Purchaser herein.

Trademarks

All other trademarks associated with the Software are trademarks owned by NDCHealth or Third-Party Licensors. No right, license, or interest to such trademarks is granted hereunder, and Purchaser agrees that no such right, license, or interest shall be asserted by Purchaser with respect to such

trademarks. Purchaser agrees not to take any action which would damage the goodwill associated with any such trademarks.

Limited Warranty

NDCHealth warrants to Purchaser that the computer media on which the original Software is recorded will be free of defects in materials and workmanship for a period of thirty (30) days from the date of purchase under normal conditions of use and service. Should the media become defective within thirty (30) days from the date of purchase, if proof of original purchase can be verified, NDCHealth will replace the Software or, at its option, NDCHealth may refund to Purchaser the original NDCHealth purchase price.

NDCHealth, its software suppliers, distributors, and Value-Added Resellers make no representations, warranties, or guarantees, express or implied, including warranties of merchantability or fitness for any use or any particular purpose with regard to the Software and documentation. The Software and documentation are licensed to Purchaser on an AS IS basis. Liability of NDCHealth will be strictly limited to the purchase price of the Software in any case.

IN NO EVENT SHALL NDCHEALTH BE LIABLE FOR ANY DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS, LOSS OF BENEFITS, OR LOSS OF DATA RESULTING FROM THE USE OF THIS SOFTWARE OR ARISING OUT OF ANY CLAIMED OR ACTUAL BREACH OF WARRANTY. Some states do not allow the exclusion or limitation of direct, incidental, or consequential damages. If Purchaser resides in such a state, the above limitation may not apply to Purchaser.

NO OTHER REPRESENTATIONS OR WARRANTIES CONCERNING THE SOFTWARE OR DOCUMENTATION MAY BE GIVEN EXCEPT THOSE CONTAINED HEREIN.

Termination of License

NDCHealth may terminate this License Agreement upon a breach by Purchaser of any term hereof. Upon such termination by NDCHealth,

Purchaser agrees to immediately return to NDCHealth or destroy all copies of the Software in Purchaser's possession.

Miscellaneous

The failure of NDCHealth to enforce any rights granted hereunder or to take any action against Purchaser in the event of a breach hereunder shall not be deemed a waiver by NDCHealth as to subsequent enforcement of rights or subsequent actions in the event of future breaches. Neither this License Agreement nor any interest in this License Agreement is assignable by Purchaser. This License Agreement represents the entire agreement concerning the Software between Purchaser and NDCHealth. Should any term of this License Agreement be declared void or unenforceable by any court of competent jurisdiction, such declaration shall have no effect on the remaining terms hereof. This License Agreement shall be construed and governed in accordance with the laws of the State of Georgia, United States of America.

Technical Support

■ Independent NDCMedisoft Value-Added Reseller

There are independent Value-Added Resellers of NDCMedisoft in your market area who are knowledgeable and efficient in selling, installing, troubleshooting and supporting your NDCMedisoft program. You can contact your NDCMedisoft Account Executive for the name of a qualified independent Value-Added Reseller in your area to give you hands-on help.

■ NDCMedisoft Technical Support

Call Toll-Free (800) 334-4006. Get help when you need it with NDCMedisoft technical support services! NDCMedisoft support is available to answer questions and assist in troubleshooting problems.

NDCMedisoft support answers questions related to the operation of NDCMedisoft software in a physician's office or a billing service. NDCMedisoft technicians are unable to assist with network configuration, computer hardware problems, or training on how to do

medical billing. NDCMedisoft support will provide software assistance to any customer, no matter where the program was purchased.

When You Call Support

You'll get faster service if you have these items ready when you call NDCMedisoft support:

- The Serial Number and registration information for your NDCMedisoft software.
- Your NDCMedisoft customer number. This is found on the upper right corner of the invoice or packing slip that came with your NDCMedisoft program.
- A complete description of your problem or question, including the complete text of any error messages.
- A major credit card if your free support period has expired.
- It is usually necessary for you to be able to work on your computer while you are talking to the technical support staff, so be sure your phone is close to the computer.

Service Hours

Remember, Arizona doesn't change to daylight saving time. Year around support is available from 6:00 AM until 5:00 PM, Monday through Friday, Mountain Standard time.

Support

Contact your independent NDCMedisoft Value-Added Reseller, NDCMedisoft sales at (800) 333-4747, or visit the NDCMedisoft web site at <http://www.medisoft.com> for details.

Support pricing is subject to change without notice.

Chapter 1

Introduction

The Advantage Database Server

Advantage Database Server is a high performance client/server RDBMS for stand-alone, networked, Internet, and mobile database applications. The Advantage Database Server allows developers the flexibility to combine powerful SQL statements and relational data access methods with the performance and control of navigational commands. Advantage has native development interfaces designed to leverage existing knowledge of popular development tools. With optimized data access methodology for easily delivering unparalleled performance, Advantage provides security, stability, and data integrity while being completely maintenance-free.

The Advantage Database Server is the key to improved database performance in network environments. The server can be visualized as an intelligent controller that reduces competition for resources and off-loads much of the work normally performed by each client workstation. It is responsible for all database access, including all reading and writing of data and lock management. Working with the network operating system, the Advantage Database Server processes data requests and returns the information to the network clients.

Support for NetWare and Windows NT/2000 Operating Systems

The Advantage Database Server supports the NetWare and Windows NT/2000.

The Advantage Database Server for NetWare is implemented as a NetWare Loadable Module (NLM). An NLM is simply an executable for the NetWare operating system. The Advantage Database Server for Windows NT/2000 operates as a Windows NT/2000 Service.

Note: The Advantage Database Server for Windows NT/2000 is a Service. It cannot be run as a standard Windows application. Refer to Chapter 2, *Installing the Advantage Database Server*, for more information on Windows NT Services and how to start them.

The Advantage Database Server retrieves requests for database operations to be performed on behalf of the clients. The Advantage Database Server locates tables on the server and processes the database operations. The result of the operation is then returned to the client across the network, eliminating the need to send the database to the client for processing. This provides far better concurrency control and system integrity than is otherwise available.

Traditional non-client/server applications send raw data from the server across the network to be processed on the workstation. With the Advantage Database Server, much of the data is processed by the Advantage Database Server on the file server. By decreasing network traffic, you increase performance.

The Advantage Database Server integrity system ensures that database updates either run to completion or do not begin. The Advantage Database Server will not execute partial commands. This means that the integrity of your database no longer depends on the stability of the workstations on the network. Because the Advantage Database Server is responsible for all database access (on behalf of the clients), it can do a far better job of concurrency control than traditional systems, where concurrency must be synchronized between remote workstations. Better concurrency control means better multi-user performance.

The Advantage Product Family

The Advantage Database Server is the database engine that supports an entire family of Advantage products. Brief descriptions of the Advantage Client Kits are provided below. For more information, reference your client-specific Advantage documentation, your Advantage Internet Server documentation, or visit the Advantage Web site at www.AdvantageDatabase.com.

Advantage Client Kits

Advantage clients can seamlessly replace existing database drivers with fully compatible Advantage drivers. Previously developed database applications can be easily converted to access the Advantage Database Server using customized Advantage development interfaces. Advantage clients are free and are included on the Advantage product CD.

Available Advantage clients include native, integrated solutions for Delphi, C++Builder, CA-Clipper, and CA-Visual Objects. Windows development platforms can also access the Advantage Database Server and Advantage Local Server via a variety of Advantage clients including (but not limited to) the Advantage Client Engine API, the Advantage ODBC Driver, the Advantage Perl DBI Driver, and the Advantage OLE DB Provider.

Chapter 2

Installing the Advantage Database Server

System Requirements

■ **Networks**

Novell NetWare 4.x

Novell NetWare 5.x or later

Microsoft Windows NT/2000

■ **Network Protocols (on the Server)**

IPX with NetWare 4 and newer, Windows NT/2000

IP with NetWare 5 and newer, Windows NT/2000

■ **File Server Processors**

Multiple processors supported with NetWare 5 and newer, Windows NT/2000

■ **Client Operating Systems**

Windows NT/2000/XP

Client database applications developed with supported Advantage clients

■ ***Client Development Platforms***

Delphi (via native components, OLE DB Provider (ADO), ODBC Driver, or API)

C++Builder (via native components, OLE DB Provider (ADO), ODBC Driver, or API)

Visual Basic (via OLE DB Provider (ADO), ODBC Driver, or API)

Access 97/2000 (via OLE DB Provider (ADO), ODBC Driver, or API)

Visual C++ (via OLE DB Provider (ADO), ODBC Driver, or API)

Visual FoxPro (via OLE DB Provider (ADO), ODBC Driver, or API)

Perl (via DBI driver)

PHP (via Advantage PHP Extension)

CA-Visual Objects (via RDD, OLE DB Provider (ADO), ODBC Driver, or API)

Clipper (via RDD)

Any development environment that can access ADO/OLE DB, an ODBC driver, or make a call into a Windows DLL or Linux shared object (via API)

■ ***Xbase Table Formats Supported***

Clipper-compatible DBF, FoxPro-compatible DBF

■ ***Xbase Index File Formats Supported***

Clipper-compatible NTX, FoxPro-compatible CDX, FoxPro-compatible IDX

■ ***Xbase Memo File Formats Supported***

Clipper-compatible DBT, FoxPro-compatible FPT

■ ***Proprietary Advantage Table Format Supported***

ADT

■ ***Proprietary Advantage Index File Format Supported***

ADI

■ ***Proprietary Advantage Memo File Format Supported***

ADM

■ ***Licensing***

The Advantage Database Server is sold on a per-server basis with User Options to provide concurrent users up to the maximum specified. Each PC workstation running one or more Advantage applications is considered a user.

Installing and Loading the Advantage NLM for NetWare

Note: You must have administrative privileges to install the Advantage Database Server.

Installation

1. Insert the Advantage Database Server install CD in the CD-ROM.
2. Proceed through the introductory windows that appear on the screen.
3. Select the version of NetWare the Advantage Database Server is to be installed upon. It is important that correct version of NetWare be selected. If "NetWare Version 5.x or Newer" is selected, the Advantage Database Server for NetWare 5 will be installed. This Advantage Database Server NLM has support for IP and multiple processors. This NLM will not load on a NetWare 3.x or 4.x server.
4. Once the Advantage Database Server files have been copied to the NetWare server, the ***Product Information*** window prompts you to enter the Advantage serial number, validation code or authorization code, and the name of the registered owner. Reference the Advantage Database Server Serial Number ID Card included with the product to find your serial number and validation code or authorization code.
5. Click [Finish] to complete the installation.

Loading the Advantage NLM

The Advantage Database Server NLM must be loaded before applications can run using the Advantage Database Server. Once the Advantage Database Server files are installed, you can load the Advantage Database Server NLM into file server memory any time the file server is running.

To load the Advantage Database Server NLM using the default configuration, enter the following from the NetWare system console:

LOAD SYS:\ADS\ADS.NLM

When the NLM is loaded, the Advantage Database Server NLM screen is displayed, and the Advantage Database Server NLM is ready to accept client commands.

Unloading the Advantage NLM

To unload the Advantage Database Server NLM, follow one of these two steps:

1. If the Advantage Database Server NLM screen is displayed, press the **(F5)** key. You will be prompted to complete the unloading of the NLM.
2. If the Advantage Database Server NLM screen is not displayed, enter the following from the system console at the ":" prompt:

UNLOAD ADS.NLM

You can unload the Advantage Database Server NLM from the file server at any time. However, if the Advantage Database Server NLM is unloaded while users are active, there is the potential for database corruption, because tables and index files may be only partially updated. Verify that there are no users active when unloading the Advantage Database Server NLM. After the NLM is unloaded, all resources are returned to the NetWare operating system.

Installing the Advantage Service for Windows NT/2000

Note: You must have administrative privileges to install the Advantage Database Server.

1. Insert the Advantage Database Server install CD in the CD-ROM.
2. Proceed through the introductory windows that appear on the screen.
3. Once the Advantage Database Server files have been copied, the *Product Information* window prompts you to enter the Advantage Database Server serial number, validation code or authorization code, and the name of the registered owner. Reference the Advantage Database Server Serial Number ID Card included with the product to find your serial number and validation code or authorization code. Radio buttons exist to choose the Advantage Database Server Service Startup option. If you are unsure of which startup option to choose, the default Automatic is recommended.
4. Click [Finish] to complete the installation.

To start the Advantage Database Server Service, you must start the **Advantage Database Server** service from the Service Control Manager. See the Starting and Stopping the Advantage Service section on page 10 for more information on how to start the Advantage Database Server for Windows NT/2000.

Understanding Windows NT/2000 Services

In Windows NT/2000, many server-side programs act as a service. Unlike regular applications, services run in the background providing application support and have no user interface of their own. Most services can be started, stopped, paused, and continued. Windows NT/2000 services are controlled through the Windows NT/2000 Service Control Manager. The Service Control Manager lists all installed services and their current status.

To access the Service Control Manager with Windows NT 4.0, open the Control Panel folder and double-click the Services icon. To access the Service Control Manager with Windows 2000, click the [Start] button, select **Programs, Administrative Tools, Services**, and finally double-click on the

Services icon. Among the information provided by the Service Control Manager is the following:

- **Server status**—relates the current status of the service (started, stopped, or paused)
- **Startup options**—allows you to select the startup type for the selected service (automatic or manual)
- **Startup parameters**—the startup parameters box allows you to specify startup parameters to a particular service

Advantage Database Server Service

The Advantage Database Server was designed as a service to provide the most robust and safest database management possible. As a service, Advantage has better control over how and when the program is started and shut down. For example, if the startup type *Automatic* was selected during installation of the Advantage Database Server, the Advantage Database Server service will automatically start when the server boots up after being shut down. This provides a benefit over regular applications because it does not require a user to log in to get the Advantage Database Server service up and running again after a power failure or other unexpected shut down.

Starting and Stopping the Advantage Service

The Advantage Database Server Service must be started before applications can run using the Advantage Database Server. Administrative privileges are required to stop a service. The pause and continue features are not available with the Advantage Database Server Service.

To start and stop the Advantage Database Server Service for Windows NT 4.0:

1. Open the Control Panel.
2. Double-click the Services icon.
3. Select the **Advantage Database Server** from the list provided.
4. Click either the [Start] or [Stop] button as required.

To start and stop the Advantage Database Server Service for Windows 2000:

1. Click the [Start] button on the taskbar.
2. Select **Programs, Administrative Tools, and Services**.
3. Select the **Advantage Database Server** from the list provided.
4. Click either the [Start] or [Stop] button as required.

Using Windows NT/2000 Event Logging

Windows NT/2000 event logging provides a centralized way for applications and services to record events such as when a service was stopped or started as well as error conditions. The Windows NT/2000 Event Viewer provides a standard user interface for viewing logged events. There are three types of event logs:

- **System Log**—tracks Windows NT/2000 system information
- **Security Log**—tracks events triggered by security violations
- **Application Log**—tracks events written by an application

The Event Viewer is accessed through the Administrative Tools group/folder. Select the type of log file you want to view (System, Security, or Application) from the Log menu option. Double-click on any event in the events list to view event details.

Event Logging and the Advantage Database Server Service

Information, warnings, and errors pertaining to the Advantage Database Server Service are recorded in the Event Application Log. Information messages are used to record when the Advantage Database Server Service was started or stopped. Advantage Database Server service warnings alert users of recoverable problems. Errors, however, are used to display non-recoverable conditions that are likely to cause a service failure.

The Advantage Database Server Service also maintains an error log for additional, non-system related Advantage Database Server errors. By default,

the error log is located in the root directory of the drive where the Advantage Database Server is installed. The location of this log is configurable.

Chapter 3

Advantage Functionality

Client/Server Technology

"Client" normally refers to workstations and applications that are attached to a network. "Server" is the central repository of files that are commonly shared by the "client" applications. True client/server technology involves an intelligent division of processing between the client and the server.

Optimizing your network resources and performance depends on a complete understanding of the relationship of "clients" to "servers." For example, certain operations are ideally suited to reside on the client; such as the user interface and most business rules and operations. Operations involving database access are most appropriately performed on the server where the data files are located. By contrast, in a non-client/server application, all database operations are executed on the client while the server simply stores the tables for sharing among others on the network.

The Advantage Database Server optimizes your client/server architecture by increasing multi-user performance, database stability, and database security while making it easy to install and integrate Advantage into your existing applications. Performance improvements are achieved primarily by reducing network traffic, intelligently maintaining database files, and by providing intelligent lock management.

Multi-User Performance

In a multi-user network environment, traffic to and from the network is nearly always the bottleneck that leads to poor multi-user performance. Thus, the key to improving performance is to reduce the amount of data transferred across the network.

Optimized Data Access

In non-client/server environments, all application and data processing takes place on the individual client workstations. The file server, where the data files are located, serves only as an unintelligent, shared hard disk. The server CPU also remains largely unused. When the database needs to be updated, all necessary table and index data is read from the server, across the network, to the client workstation. Actual updates occur on the client workstation. The new data is then sent back across the network where it is written to the file server. When a piece of information needs to be found in the database, a search of the database is required. Index data must be read from the server, across the network, to the client workstation where the search of the data takes place. Index data must be continually read over to the client until the desired data is found or until it is determined the data does not exist in the database. As a result, non-client/server database applications typically suffer from poor performance. As more users are added to the system, the larger the database becomes, the amount of data transferred across the network increases even more, and performance deteriorates even further.

With the Advantage Database Server, network database application performance is increased dramatically. Advantage uses the client/server architecture to off-load all index traffic associated with index look-ups and index maintenance. The Advantage Database Server allows a central point of control for all database operations. No index data is ever transferred across the network to the client workstation. All database opens, reads, and writes are performed on the server by the Advantage Database Server where the data exists. Since no index data and no unnecessary table data is ever sent to the client workstation, network traffic is drastically reduced and performance increases. An additional benefit of Advantage client/server technology is that the Advantage Database Server performs all data processing on the file server, therefore the CPU on the file server is fully utilized. Advantage database applications have both the workstation CPU and the file server CPU working for them.

Intelligent Lock Management

Another critical, but often overlooked, factor contributing to increased performance with Advantage client/server technology is data locking that occurs behind the scenes by the database system. Before index data can be read or updated, an implicit lock is obtained on that index. As an application developer, you have no control over when or how these implicit locks are obtained. The database system that the application is using obtains these locks for you.

In non-client/server environments, all internal locking requests are issued from the individual client workstations. In multi-user systems, there is much contention for index access. The initial attempts to lock the index files will often fail because another user already has the index locked. Non-client/server database systems try and retry the locks from the client, which takes more time and leads to poor performance for that application. In addition, these lock retries generate increased network traffic that will slow performance for all users on the network. Non-client/server application index locking generally leads to slow performance for that individual application as well as compounding performance problems for other applications on the network.

With Advantage client/server technology, the internal index locks occur on the server. The Advantage Database Server uses an intelligent lock management system with its proprietary locking mode that eliminates lock retries and requires no network traffic. The Advantage Database Server uses an internal queuing algorithm that allows for read-through index locking and immediate index write locking. In non-client/server systems, there is no differentiation between index "read" locks and index "write" locks. There is only one kind of lock that can be obtained whether reading or updating an index, and thus all index access is sequentialized. With the Advantage Database Server, index "read" locks exist as well as index "write" locks. When an index is being read, there is no danger of that reader changing the structure of that index. Thus, Advantage allows read-through index locking for index "read" locks. This means all users who desire to read the same index can get concurrent "read" locks on that index and read from the index all at the same time. As you can imagine, this increases multi-user index "read" performance immensely.

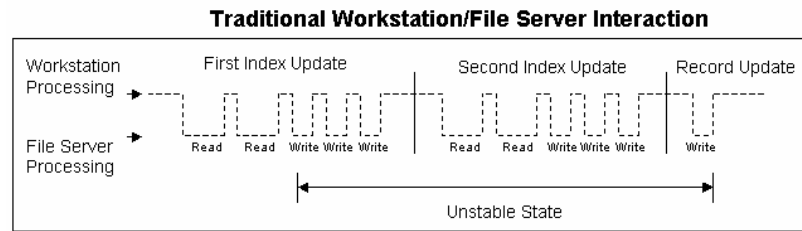
When an index is being updated, it is possible that the structure of the index may change. Therefore, when a user obtains a "write" lock on an index, no other users can obtain a "read" or "write" lock on the index. The Advantage queuing process allows for users to "line up" for access to that index. As soon

as the update operation completes and the "write" lock on the index is released, the next user in the queue will immediately obtain his index lock and can proceed with the index read or write operation. No lock retries are ever required. "Write" lock requests are simply queued, and the locks are immediately granted as soon as the prior user has released the lock. This "write" lock queuing and the elimination of lock retries significantly increases multi-user database application performance.

Database Stability

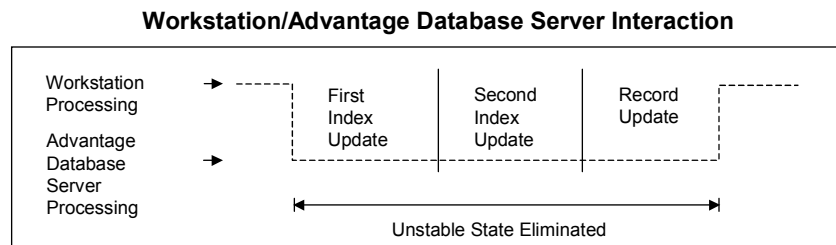
During traditional non-client/server interaction between a workstation and server, tables and index files are susceptible to corruption. Workstations can be interrupted or fail because of a reboot, power failure, or memory problem. It takes several calls between the workstation and the server to complete an update operation. If during this process the application, workstation, or network fails, the operation is partially executed, leaving the database in an unknown state. Index file stability and possibly table stability are compromised. The following diagram shows the interaction between client and server in a non-client/server environment during an update to a single record in a table that causes two related indexes to be updated. Note that if the application, workstation, or network fails at any time between the first index write and the write of the table record, the database will be left in an unstable and corrupt state.

Note: Due to space considerations, the example in the diagram below has been overly simplified. In fact, many more read and write operations than are shown must occur between the workstation and file server before each index update can be completed. In actuality, there is even a larger window for possible database corruption in non-client/server environments than shown.



Advantage provides database stability ensuring that every database operation is executed completely or is not executed at all. Entire database update operations are executed on the server. Therefore, if the application, workstation, or network fails, the database operation will either successfully be transmitted to the Advantage Database Server or not transmitted at all. The status of the application, workstation, and network cannot affect the data in your database. By transmitting entire table and index file update operations in one command from the client to the server, Advantage eliminates corruption errors introduced by application, workstation, or network failure.

The following diagram shows the interaction between client and server with Advantage client/server processing during the same update shown above to a single record in a table which causes two related indexes to be updated. Note that there is no unstable state with Advantage as no database updates will occur until all required update information has been received by the Advantage Database Server. Once all update information has been received, those updates will occur on the server by the Advantage Database Server requiring no network traffic and no workstation involvement.



Chapter 4

Advantage Database Server Features

Database Security

Database security is essential for controlling access to the files in your database. If database security is not present, there is little or no control over whom can update data, delete files, and/or possibly corrupt data in the database. The Advantage Database Server provides two methods of database security for free connections:

- **Check Rights:** Checking the user's network access rights before opening files for that user.
- **Ignore Rights:** Allowing access to the database via an Advantage application only.

Check Rights

When a non-Advantage application, whether it be a database application or other application, needs to open or create a file on the file server, the network operating system will verify that the user has sufficient network access rights to that directory and/or file before allowing that user to open or create the file. If the user has no network access rights to the directory and/or file, the network operating system will not allow the application to open or create the file. If the user has limited network access rights to the directory and/or file, such as read-

only access, the network operating system will only allow the application to open a file for read-only use.

The Advantage Database Server "Check Rights" security method for free connections obeys this same concept of verifying user network access rights. The Advantage Database Server will verify the user's network access rights to a directory and/or file before opening or creating the specified file for the user. The data is "secure" because only those users with sufficient network access rights can open and/or modify the data files.

Ignore Rights

Restricting a user's network access rights to the directory and/or files as described in the Check Rights security method often does not provide enough database security for free connections. If a user has been given the necessary read, write, create, and/or delete rights necessary to read, write, create, and/or delete data via your database application, then that user can also read, write, create, and/or delete data without using your application. Users can maliciously or accidentally corrupt the database by writing to the database with uncontrolled database editors. Files in the database could also be purposely or accidentally deleted entirely. What mission critical database applications often need is an additional level of security that only allows users to access the database via your database application. That way the database application has full control over what users are reading, writing, creating, and/or deleting data in the database. Non-client/server applications have no way to enforce this additional security; the Advantage Database Server does.

The Advantage Database Server "Ignore Rights" security method for free connections allows you to "hide" files in the database from all users who are not accessing data through an Advantage application.

The first step necessary to provide the Advantage "Ignore Rights" method of security is to have the system administrator remove network access rights from all users who could potentially damage the database. Once network access rights have been revoked from users to the database directory and/or files, users cannot maliciously or accidentally corrupt the database by writing to the database, creating new files, or deleting existing files in the database because they no longer have access to those files.

The second step necessary to provide the Advantage "Ignore Rights" method of security is use the "Ignore Rights" method of Advantage rights checking in

your application when opening and creating files for free connections. When a file is to be opened or created by the Advantage Database Server, and "Ignore Rights" security is being used, the Advantage Database Server will NOT verify that the user has sufficient network access to the directory and/or file and will open or create the file for the application regardless of the user's network access rights. The Advantage Database Server can do this because it is running on the server and is running at a "supervisor" level. Using Advantage's "ignore rights" security method allows your Advantage application to have full control over who can access the database and how the database can be modified. Only Advantage applications using the "ignore rights" security method may access the database. Non-Advantage applications will have no access to the database at all.

The first step described above, in which the system administrator should remove network access rights from all users who could potentially damage the database, is very flexible. You and/or the system administrator can decide if all users and/or groups will have their network access rights revoked or if just certain users and/or groups will have their rights revoked. You and/or the system administrator can also decide if all network rights will be revoked, or if just some rights (such as delete and write rights) will be revoked. For example, if a report writer is being used, but that report writer is not an Advantage application, that report writer will be unable to function if all rights have been removed from all users. If a network "REPORTS" group is created and that group is given read-only access to the database directory and/or files, then users who are members of the "REPORTS" group can run the report writer utility in read-only mode against the database.

Note: Ignore Rights security is not available with the Advantage Local Server. Check Rights security is effectively used during any connection to the Advantage Local Server.

Note: If you are using the Advantage Database Server for Windows NT/2000, and your data files are located on a drive using NTFS, that PC's "system" group must have full access to the share that contains the data in order for the Advantage Database Server service to have access to that data. Note that it must be that PC's system group that has full access, not the domain's system group.

Configured Semaphore Connection File Directory Often Necessary with the Advantage Database Server

When an Advantage application either calls an Advantage "connect" API (if available) or opens the first table, it establishes a connection between the application and the Advantage Database Server. During the connect, a semaphore connection file is created by the Advantage Database Server if the Use Semaphore Files configuration setting is set to 1. The semaphore connection file is implicitly opened by the Advantage application using a generic file open call. That is, the file is not opened through the Advantage Database Server. The semaphore connection file is used to aid in determining the connection status between the workstation and the server. The default directory in which this semaphore connection file is opened is the server directory specified in the "connect" API or where the first table is to be opened. The directory in which semaphore connection files are opened is configurable via the Advantage configuration file, ADS.CFG. Since the semaphore connection file is actually opened from the workstation and not via the Advantage Database Server, the user must have network READ rights in the directory in which the semaphore connection file exists.

When using the Advantage "Ignore Rights" method of security, user access rights are usually revoked from the directory where the data exists and, thus, where the semaphore connection file is created and needs to be opened. Therefore, it usually makes sense to configure a specific semaphore connection file directory in the Advantage Configuration file where no important data exists, but where all users have been given at least network READ access.

The network administrator can take further advantage of this semaphore connection file creation if the Use Semaphore Files configuration setting is set to 1 to limit which users can connect to the Advantage Database Server and have access to the database. Each user's network rights to the configured semaphore connection file directory will determine if a user can connect to the Advantage Database Server. Those users having network READ rights in the semaphore connection file directory will be able to connect to the Advantage Database Server. Those users who do not have network READ rights will not be able to connect to the Advantage Database Server and thus, will not be able to open any data files.

Note: Opening the semaphore connection file does not obey the Advantage "rights checking" setting. A user must have at least READ access rights to the directory where the semaphore connection files

are created in order to connect to the Advantage Database Server no matter how the "rights checking" setting is set.

Note: The information above is only pertinent if you are connecting to Advantage with 16-bit client applications, or if you explicitly set the Use_Semaphore_Files option on the server. See the configuration option Use_Semaphore_Files in the Advantage Database Server Help documentation (ADS.HLP) for more information.

Advantage Locking Modes

Locking records and files is critical to sharing data on a network. Locking operations help enforce database stability by preventing other users from accessing the data while it is being updated. In non-client/server environments, lock management operations occur from the client and can be quite slow. The Advantage Database Server provides data locking from the server which significantly increases application locking management and performance. The Advantage Database Server provides an intelligent, high performance Proprietary locking mode and a Compatibility locking mode for sharing data with non-Advantage applications.

The Advantage locking method used by the Advantage Database Server can be specified per table.

Advantage Proprietary Locking

The Advantage Database Server uses an intelligent lock management system with its proprietary locking mode that eliminates lock retries and requires no network traffic. The Advantage Database Server uses an internal queuing algorithm that allows application locks to occur without making network operating system lock API calls. All locking information is maintained internally in the lock queues.

Advantage Proprietary locking mode also allows for read-through index locking and immediate index write locking. When an index is being read, there is no danger of that reader changing the structure of that index. Thus, Advantage allows read-through index locking for index "read" locks. All users who desire to read the same index can get concurrent "read" locks on that index and read from the index all at the same time. As you can imagine, this

increases multi-user index "read" performance immensely. When an index is being updated, it is possible that the structure of the index may change. Thus, when a user obtains a "write" lock on an index, no other users can obtain a "read" or "write" lock on the index. The Advantage queuing process allows users to "line up" for access to that index. As soon as the user is done updating the index and releases the "write" lock on the index, the next user in the queue will immediately obtain an index lock and can proceed with the index read or write operation. No lock retries are ever required. "Write" lock requests are simply queued, and the locks are immediately granted as soon as the prior user has released the lock. This "write" lock queuing and elimination of lock retries increases multi-user database application performance.

Advantage Proprietary locking is only available with the Advantage Database Server. Advantage Proprietary locking is not available with the Advantage Local Server.

When Advantage Proprietary Locking is used, files are first opened in a "deny write" mode to non-Advantage users. Since the files cannot be opened in a writable mode by non-Advantage users, the Advantage Database Server can assume the environment is Advantage-only and internally maintains specific locking information. Any non-Advantage application can open files in a shared, read-only mode only. Likewise, the Advantage Database Server cannot open files that were opened by some other application in a writable mode.

When Advantage Proprietary Locking is used on Advantage ADT files, index "write" locking can be performed per tag. Therefore, multiple applications can be updating different tags in the same ADI file at the same time. This per tag "write" locking is not available with Xbase index files.

Advantage Compatibility Locking with Xbase Files

With Xbase files, the Advantage Compatibility Locking mode is provided to allow data to be shared by Advantage applications and non-Advantage applications simultaneously. When using Advantage Compatibility Locking, the Advantage Database Server cannot make full use of the internal queuing algorithm for lock management used with Advantage Proprietary Locking. Read-through index locking is not available. Additionally, locks must be made visible to non-Advantage applications by obtaining network operating system locks. When using Compatibility Locking, files are opened in the mode specified by the application. That is, if the file is specified to be opened in a shared mode, the file is opened by the Advantage Database Server in a "deny

none" mode. This also allows non-Advantage applications to open the tables and index files in a read/write mode.

With Xbase files, Advantage Compatibility Locking is available with both the Advantage Database Server and the Advantage Local Server.

Note: Advantage Compatibility Locking allows other applications to write to tables and index files. This decreases the amount of concurrency Advantage can provide and reduces performance. There is a possibility of index corruption with Advantage Compatibility Locking because tables and index files may become only partially updated if a workstation goes down that is running a non-Advantage application. It is recommended that you use Advantage Compatibility Locking only when first converting your applications to use Advantage which share files with other non-Advantage applications. Once all applications are converted to use Advantage, use Advantage Proprietary Locking to regain index integrity and concurrency control, and improve performance.

Advantage Compatibility Locking with Advantage ADT Files

With Advantage ADT files, the Advantage Compatibility Locking mode allows data to be shared by Advantage Local Server applications simultaneously. Locks in one Advantage Local Server application are made visible to other Advantage Local Server applications by obtaining network operating system locks. Files are opened in the mode specified by the application. Therefore, multiple Advantage Local Server applications can have the same files open at the same time in a read/write mode.

With ADT files, Advantage Compatibility locking is not available with the Advantage Database Server. Advantage Compatibility locking is only available with the Advantage Local Server.

Single Application Scalability

An application is scalable if only the single application is needed to access data in various environments. Advantage provides many scalability features and functions that are accessible from different development platforms.

With Advantage, functions and/or drivers are provided that allow a single application to open files that exist on the client workstation as well as files that exist on a file server. A single Advantage application can access data on a Novell NetWare server using the Advantage Database Server for NetWare, a Windows NT/2000 server using the Advantage Database Server for Windows NT/2000, or a Linux server using the Advantage Database Server for Linux. The same single application can also communicate to the Advantage Database Server via either the IPX protocol or IP protocol, depending on which is available and supported.

Reduced File Handle Usage

The slowest operation that can be requested of an application is to open a file, especially in a network environment. Given this fact, writing fast database applications usually requires that once a table is opened, it remains open. However, complex database applications often require access to many files. This is a problem because database operating systems have a limited number of files that can be open at one time. Non-client/server application developers traditionally have had to open and close files as they are needed, unnecessarily slowing the application down with the opening overhead.

The Advantage Database Server eliminates this problem by providing a single point of access to the database. The Advantage Database Server opens data files and provides access to users. The need for local file handles is eliminated. The need for server file handles is eliminated because the Advantage Database Server does not use any file handles to access a table shared among multiple users. For example, suppose you had an inventory control program with 25 users. If the program uses 16 tables, with an average of three index files each, it could require up to 64 file handles (i.e., 16 table handles + (3 index file handles X 16 tables) = 64 file handles being used). To keep all the files open, a typical non-client/server application would need 64 file handles at each workstation and 1600 file handles at the file server. The same application using the Advantage Database Server would require no file handles per workstation and only 64 file handles at the file server.

Advantage Optimized Filters

Advantage Optimized Filters (AOFs) provide high performance filter optimization for Advantage applications. AOFs speed filter processing by using index keys instead of table records. If a specified field has an index built on it, an AOF uses the index file rather than the table to process the filter. AOFs are used to increase performance of record filtering when directly opening a table for navigational use and setting a filter on that table, as well as used to increase the performance of filtering records in the WHERE clause of an SQL query. Reducing the amount of data that must be retrieved from the disk increases performance.

An AOF can be thought of as a query on the table. When an application sets an AOF on a table opened directly, or issues an SQL SELECT query with a WHERE clause, the Advantage Database Server will build an AOF to filter those records. The server uses indexes that have been opened for the given table to quickly determine which records are in the AOF. The actual AOF consists of an array of bits (in natural record number order) where each bit represents a single record. The bit for each record that passes the filter condition is turned on.

In general, AOFs are created by matching portions of the filter or WHERE clause expression with index key expressions. For example, if the field "lastname" is indexed, Advantage can quickly optimize the filter expression "lastname >= 'W' .AND. lastname <= 'Wil'". The Advantage Database Server will Seek to the appropriate location in the index and traverse the index pages and set bits in the AOF for records that pass the filter condition. When doing this, Advantage does not read the actual records but will read only the necessary index pages needed to resolve the filter.

AOFs are created when an SQL SELECT query has a WHERE clause or by one of the following methods when opening a table directly:

- Calling the AdsSetAOF function with the Advantage Client Engine API
- By default, the Filter property in the TAdsTable and TAdsQuery components will create an Advantage Optimized Filter. To use traditional record filters with the Filter property rather than AOFs, set the AdsOptimizedFilters subproperty to False
- By default, all filters set will use AOFs with the Advantage CA-Visual Objects RDDs. Calling the Visual Objects function RDDSetInfo(

`_SET_OPTIMIZE)` or the command "Set Optimize OFF" provides the ability to disable the use of AOFs

AOFs are only optimized for DBF tables with CDX index files and with ADT tables with ADI index files. AOFs are not fully supported with DBF tables that use NTX indexes. If an AOF is built on a DBF table with an NTX index, the AOF will be non-optimized. Under certain circumstances, it may still be advantageous to use AOFs with DBF tables and NTX indexes. If multiple passes are to be made through the table with the AOF set (e.g., due to data displayed in a grid), the AOF will generally be faster because all records that do not pass the filter condition will be removed from the AOF during the first pass through the data, and they will not be read during subsequent passes.

Because the Advantage Database Server creates and uses AOFs, network traffic is eliminated when creating AOFs on the Advantage Database Server. With Advantage Local Server, however, network traffic is an issue when creating AOFs on tables that reside on network drives because the index data used to create the AOF must be transported across the network to the client where the Advantage Local Server will create the AOF bitmap. But AOFs with Advantage Local Server should still provide much better performance than when using traditional record filters because only those records that pass the filter condition will ever need to be read over to the client PC.

When using AOFs with the Advantage Database Server, there will be some increased resource usage on the server. This is especially true if non-optimized filters are created. For example, if the filter "lastname = 'Smith'" is used and there is no index available on lastname, the filter will not be optimized. This means that every bit will be turned on initially. Therefore, the memory allocation will be approximately $\text{RecordCount} / 8$ bytes. The total memory allocation for a 1,000,000 record table will be approximately 125,000 bytes. As a developer, you should keep this in mind when using AOFs. The allocation is probably "small" for most server configurations, but if you have a very large number of users forcing allocations like this, the total may be prohibitive. The memory allocation for the array of bits is done in non-contiguous blocks on a demand basis, so if a huge table has only a few records that pass the filter condition, the actual amount of allocated memory is usually quite small.

Differences Between AOFs and Traditional Record Filters

There are two primary differences between AOFs and traditional record filters. The first is the speed. When using an AOF, it is usually very fast because records that do not pass the filter condition do not have to be read from the disk. Traditional record filters, on the other hand, must always read each record to determine if it passes the filter condition.

The other difference between AOFs and traditional record filters is the freshness of the data when using the Advantage Local Server. Traditional record filters are evaluated each time a record is requested. Therefore, they always provide an up-to-date view of the data. Once a fully optimized AOF has been built, it contains the bookmarks of the records that are in the filter. In general, the Advantage Database Server keeps the filter up-to-date whenever changes are made to the records in the table. There are two cases, however, where the filter can become out of date with respect to the actual data in the tables. The first is with Advantage Local Server. Any record change made by an application will cause all AOFs associated with that table within the application to be updated, but the change will not be propagated to other applications using the same table. The second case where an AOF could be out of date is with Advantage Database Server when using DBF tables and using compatibility locking mode. In that case, it would be possible for a third-party application to make a record update to a DBF that would not be detected by Advantage Database Server. Otherwise, AOFs and traditional record filters will be consistent with each other.

Relative speeds between the two types of filters depend on the number of records in a table, the number of records that pass a filter condition, the optimization level of the filter, and server speed. It is common, though, for fully optimized AOFs to return data sets 10 to 100 times faster than traditional record filters. The largest relative speed difference between the two types of filters will be seen when a small percentage of records pass the filter condition. For example, if a filter condition passes 1% of the records in a large table, a fully optimized AOF may be 10 times faster than a traditional record filter. With a filter condition that passes 99% of the records in a table, a fully optimized AOF will be only slightly faster than a traditional record filter.

AOF Optimization

Three levels of optimization are available for AOFs. AOFs can be fully optimized (ADS_OPTIMIZED_FULL), partially optimized (ADS_OPTIMIZED_PART), or not optimized (ADS_OPTIMIZED_NONE). For example, the filter expression "lastname = 'Smith' .AND. firstname = 'John'" will be fully optimized if indexes exist on "lastname" and "firstname". If an index exists on only one of those fields, the filter will be partially optimized. If no index exists for either field, the filter will not be optimized. The following rules apply when using logical operators. The columns labeled Optimization Level refer to a simple expression (e.g., "lastname = 'Smith'") or multiple simple expressions joined by logical operators (e.g., "lastname = 'Smith'" .AND. firstname = 'John'").

Optimization Level	Operator	Optimization Level	Resulting Optimization Level
Full	.AND.	Full	Full
Full	.AND.	Not Optimized	Partial
Full	.AND.	Partial	Partial
Partial	.AND.	Partial	Partial
Partial	.AND.	Not Optimized	Partial
Not Optimized	.AND.	Not Optimized	Not Optimized
Full	.OR.	Full	Full
Full	.OR.	Not Optimized	Not Optimized
Full	.OR.	Partial	Partial
Partial	.OR.	Partial	Partial
Partial	.OR.	Not Optimized	Not Optimized
Not Optimized	.OR.	Not Optimized	Not Optimized
	.NOT.	Full	Full
	.NOT.	Partial	Not optimized
	.NOT.	Not Optimized	Not Optimized

For example, consider the filter expression:

```
hiredate > CTOD('1/1/90') .OR. ( hiredate > CTOD('1/1/85') .AND.  
deptnum = 15)
```

If field “hiredate” is indexed and field “deptnum” is not, then the expression will be partially optimized. Each simple expression involving field “hiredate” will be fully optimized but the simple expression involving field “deptnum” will not be optimized. The expression can be viewed as

Full .OR. (Full .AND. Not Optimized)

This then resolves to:

Full .OR. Partial

The final result is partial optimization.

For a simple expression to be fully optimized, the left operand of the expression must, in general, match an existing index expression exactly. For example, if the only index on a table is "UPPER(lastname)", the AOF expression "lastname = 'Smith'" will not be optimized, but "UPPER(lastname)='SMITH'" will be fully optimized. A few cases exist where the left operand does not have to match exactly. One case is for concatenated index expressions. The index expression "lastname + firstname" will be used to fully optimized the filter "lastname = 'Smith'", but it will not be used for the filter "firstname = 'John'". The following are other cases where an exact match is not needed:

- If the function YEAR(fldname) is used in an AOF expression, Advantage will use indexes of the form "fldname" or "DTOS(fldname)" to optimize the expression. For example, the filter "YEAR(doh)=1990" will be optimized if either an index of the form "doh" or "DTOS(doh)" exists
- If the function EMPTY(fldname) is used as an AOF expression, Advantage will use indexes built on “fldname” to optimize the expression. For example, the filter “EMPTY(lastname)” will be optimized if the index “lastname” exists
- If the function UPPER(fldname) is used as the index expression, Advantage will fully optimize filter expressions of the form “fldname = 'VALUE'” when the quoted text is all upper case. The filter “fldname = 'value'” would not be optimized because ‘value’ is lowercase

When a filter is partially optimized or not optimized, each bit that is set in the AOF indicates that the corresponding record may or may not be in the filter. Each bit that is cleared indicates that the record is definitely not in the AOF. Therefore, for each bit set for a partial or non-optimized filter, Advantage must read the actual record and evaluate some portion of the filter expression against the record to determine whether or not the record passes the AOF condition.

AOF Relational Operators

Advantage Optimized Filters support the following relational operators.

Operator	Description
=	Equal to
==	Strict equality for string comparison
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
#, !=, <>	Not equal to
\$	String containment

For the string containment operator (\$) to be optimized, it must have the field name on the left side of the operator just as with the other operators. For example, the filter expression "'x' \$ lastname", which will find all records that have the character 'x' in the field "lastname", will not be optimized. The containment operator can be used to efficiently find all records that have one of a set of specific characters in a specific location. For example, if the index "SUBSTR(category, 3, 1)" existed, then the filter expression "SUBSTR(category, 3, 1) \$ 'aAbB'" would be fully optimized and would find all records that have either an 'A', 'a', 'B', or 'b' in the third character of field "category". It can also be used for longer substrings, but it would generally be more efficient to use several simple expressions joined with the logical .OR. operator.

AOF Performance Tips

The amount of time required to build the Advantage Optimized Filter depends on the number of index pages that must be read. For example, the filter "lastname = 'Elliot'" would likely be built very quickly because very few (possibly only one) index pages would need to be read. On the other hand, the filter "lastname != 'Elliot'" would take longer to build because most of the index pages would need to be read. By using the logical NOT operator, a developer can speed up the AOF builds. For example, the filter ".NOT. (lastname = 'Elliot')" would build faster than "lastname != 'Elliot'" because the server would only read the index pages containing 'Elliot' and then invert the resulting array of bits. Another example is "lastname > 'C'" versus ".NOT. (lastname <= 'C')". Assuming an even distribution of last names beginning with each letter, the latter filter will build faster because the server would read fewer index pages.

With DBF tables, developers can speed up the filtering of deleted records through the use of indexes built on "DELETED()". If a table has the index "DELETED()" built, then the AOF engine will automatically use that index to optimize the filtering of deleted records. For example, consider the following two operations using the Advantage Client Engine API:

```
ulRetCode = AdsShowDeleted( False );  
  
ulRetCode = AdsSetAOF( hTable, "lastname='Smith'",  
ADS_RESOLVE_DYNAMIC );
```

If a DBF table has an index built on "DELETED()", the AOF engine will automatically add "AND !DELETED()" to the filter expression, which will cause deleted records to be removed from the AOF. If there is not an index built on "DELETED()", "AND !DELETED()" is not added to the index expression, so deleted records will still be in the AOF. The server will filter out the deleted records resulting in the same set of records that would occur with an index on "DELETED()" but there may be a loss in performance. Note that if you change the "show deleted records" setting after building the AOF on a DBF table, you should rebuild the AOF, otherwise the AOF will still filter the deleted records. With ADT tables, deleted records are not visible to the application, so there is no optimization of deleted records necessary.

The AOF engine does not use custom indexes for optimization. In general, it also does not use conditional indexes. When filtering deleted records in DBF tables, the AOF engine will use conditional indexes with conditions of

"!DELETED()". With DBF tables, unique indexes are not used for optimization. With ADT files, unique indexes are used for optimization because Advantage enforces the unique attribute and does not allow an ADT to contain records with duplicate keys for indexes with the unique attribute.

For an expression to be optimized, the left-hand side of the operator must match an index expression. If it is on the right side of the operator, it will not be optimized. For example, "Smith'=lastname" will not be optimized. This should be coded as "lastname='Smith'". In addition, the right hand side of the operator cannot vary by record. For example, the filter "lastname<firstname" will not be optimized because the right hand side of the operator contains a field value. You can get around this limitation by creating an index of the form "(lastname<firstname)". A filter of the form "(lastname<firstname)=.T." will then be fully optimized.

See your Advantage client-specific documentation for more information about Advantage Optimized Filter APIs.

Read-Ahead Record Caching

Whenever you perform a Skip (Next/Prior/MoveNext/MovePrevious) operation after performing any other movement operation in your application, the next N number of records (rather than just a single record) will be read and transferred to the client. This allows the next N - 1 subsequent Skip (Next/Prior/MoveNext/MovePrevious) operations to be performed locally. The record positioned to will be retrieved from client memory, rather than having to be read over the network from the server. This should improve Skip (Next/Prior/MoveNext/MovePrevious) performance, especially when performing such operations as populating a grid or a browse window.

When using Advantage Database (remote) Server, the default number of records read and cached on the client is the lesser of 10 or the number of records that can fit in one transmission burst, which defaults to 8K for IPX and 22K for IP. When using Advantage Local Server, the default is 10 records. To change the cache size for a given table, use one of the following mechanisms:

- AdsCacheRecords for the Advantage Client Engine API
- TAdsTable/TAdsQuery.AdsTableOptions.AdsRecordCache for the Advantage TDataSet Descendant

- Forward-only cursors for the Advantage OLE DB provider. Set the DBPROP_CANSCROLLBACKWARDS and DBPROP_CANFETCHBACKWARDS properties to FALSE. When using ADO, specify adOpenForwardOnly as the cursor type
- Forward-only cursors for the Advantage ODBC driver. Use SQL_CURSOR_FORWARD_ONLY for the SQL_ATTR_CURSOR_TYPE statement attribute
- AdsCacheRecords for the Advantage CA-Visual Objects RDDs installed from the ACE.AEF file

Note that any records cached on the client will not reflect changes made by other users to those records in the table on the server. To "dump" the cache of records currently in memory on the client and to refresh the current record, call one of the following functions which forces the client record cache to be purged:

- AdsRefreshRecord with the Advantage Client Engine API
- TAdsTable.AdsRefreshRecord with the Advantage TDataSet Descendant
- IRowsetRefresh::RefreshVisibleData with the Advantage OLE DB Provider or the Resync method if using ADO
- SKIP 0, GOTO RECNO() or AdsRefreshRecord with the Advantage CA-Visual Objects RDDs

Note that doing any movement operation other than a Skip (Next/Prior/MoveNext/MovePrevious) will cause the record cache to get dumped. Locking or updating a record will also cause the record cache to be dumped and the latest version of the record to be read to the client.

In theory, the ideal value for the number of records to read-ahead would be the number of records that normally appear in your application's grid or browse window. For example, if your application's grid or browse window contains 20 records, you may consider changing the read-ahead record value to 20 so that the entire set of records can be read with one server request. However, if your application is repeatedly doing a single Skip (Next/Prior/MoveNext/MovePrevious) operation, reading ahead 20 records and not using them may degrade your application's performance. You may want to experiment with the number of records to read-ahead in your application to find a value that provides best performance.

If your application is stepping through a recordset and updating most or all records, it will probably be best to turn off the read-ahead caching because each time a record is updated, the cache will be dumped.

See your Advantage client-specific documentation for more information about Advantage Read-Ahead Record Caching APIs.

Index

A

ADS.NLM	8
Advantage Client Kits	3
Advantage Compatibility	
Locking	
Xbase Files	24
Advantage Compatibility	
Locking ADT Files	25
Advantage Database Server	1
Installation	9
Advantage Locking Modes	23
Advantage NLM	
Loading	8
Unloading	8
Advantage Optimized Filters...	27
Advantage Product Family	2
Advantage Proprietary Index File	
Formats Supported	6
Advantage Proprietary Locking	
.....	23
Advantage Proprietary Memo	
File Formats Supported	7
Advantage Proprietary Tables	
Supported	6
Advantage Web Site.....	3
AOF Optimization.....	30

AOF Performance Tips	33
AOF Relational Operators.....	32

C

Check Rights	19
Client/Server Technology.....	13
Clients.....	5

D

Database Integrity.....	16
Database Security	19
Database Stability	16
Disclaimer	i, iv

F

File Server	5
-------------------	---

I

Ignore Rights	20
Installation	
NetWare	7
Windows NT/2000	9
Installing the Advantage	
Database Server	5
Integrity	16

Intelligent Lock Management. 15

L

License Agreement	i
Disclaimer	i
Limited Warranty	iii
Miscellaneous	iv
NDCHealth's rights	ii
Permitted Uses	i
Prohibited uses	ii
Termination of License	iii
Trademarks	ii
Licensing	7
Limited Warranty	iii
Loading the Advantage NLM ...	8

N

NDCHealth's rights	ii
NetWare Installation	7
Network Protocols	5
Networks Supported	5

O

Operating Systems	2
Optimized Data Access	14

P

Performance	
Multi-User	14
Permitted uses	i, ii, iii
Prohibited uses	ii

R

Read-Ahead Record Caching .	34
Record Caching	34
Reduced File Handle Usage....	26

S

Semaphore Connection File	22
Single Application Scalability	25
Stability	16
Starting Advantage Service	
Windows 2000	11
Windows NT	10
Stopping Advantage Service	
Windows 2000	11
Windows NT	10
Support Options	
Independent NDCMedisoft	
Dealer	iv
Technical Support	iv
Systems Requirements	5

T

Technical Support	iv
Termination of License	iii
Trademarks	ii

U

UNLOAD NLM	8
Unloading the Advantage NLM	8

W

Where to Find Help	iv
Windows 2000	
Starting and Stopping	
Advantage Service	11
Windows NT	
Starting and Stopping	
Advantage Service	10
Windows NT/2000	
Event Logging	11
Services	9
Windows NT/2000 Installation.	9

X

Xbase Index File Formats

Supported6

Xbase Memo File Formats

Supported6

Xbase Tables Supported6