OCTOBER 1991

Database Installation and Maintenance: Installing Advanced Revelation on a NetWare LAN

Rich Lee
Senior Consultant
Systems Engineering Department

This AppNote covers installation and maintenance concerns for LAN administrators who find themselves as LAN database administrators, and for database administrators looking for information about installing and maintaining server-based database applications and development packages. As a practical application of these issues, this AppNote also discusses the installation and configuration of Revelation Technologies' Advanced Revelation database product on a NetWare LAN.

Disclaimer

Novell, Inc. makes no representations or warranties with respect to the contents or use of these Application Notes (AppNotes) or of any of the third-party products discussed in the AppNotes. Novell reserves the right to revise these AppNotes and to make changes in their content at any time, without obligation to notify any person or entity of such revisions or changes. These AppNotes do not constitute an endorsement of the third-party product or products that were tested. Configuration(s) tested or described may or may not be the only available solution. Any test is not a determination of product quality or correctness, nor does it ensure compliance with any federal, state, or local requirements. Novell does not warranty products except as stated in applicable Novell product warranties or license agreements.

Contents

For more information about using Advanced Revelation on a NetWare network, contact:

Revelation Technologies, Inc.
3633 136th Place SE
Bellevue, WA 98006
(206) 643-9898

Introduction

This AppNote explores the issues involved in installing and maintaining a suitable database environment on a NetWare network. After discussing some general considerations that apply to all database programs, we will turn our attention to the first in a series of database product reviews: Advanced Revelation from Revelation Technologies, Inc. (RTI).

This material applies as much to those who have installed a database on the network as to those who have not. As we review some of the different database programs available for networking, experienced database administrators may find a few structures, attributes, and features to apply to their own database deployment and implementation. Developers, too, may see workable database implementation designs. Whatever your endeavor—installation, development tool selection, application design, or training—we hope these database reviews will be of benefit.

In preparing this AppNote, we had to review several general considerations about installing and maintaining network database systems. Some of the issues discussed (such as installation and directory structures) were tested in our laboratory; other items have been submitted by consultants both internal and external to our organization. We appreciate their interest, and hope to include relevant material from outside whenever possible in future AppNotes.

Of our major concerns, the most all-encompassing one goes beyond just getting whatever database or databases you decide on installed successfully. You must also be able to keep them operational over the long haul. Many database product reviews in our industry concentrate on performance (for example, is the transaction performance up to snuff), with little or no mention of development and maintenance aspects. In this AppNote, our direction will be to hold off on performance reviews and pursue suitable examples of network implementation that reflect good network database management principles. The intention is to devote most of our time discussing what it takes to install and maintain database systems and applications in a LAN environment.

In order to do that, we will focus on these aspects of database implementation:

- Installation and documentation
- Maintenance and directory structures
- Development and production concerns
- Interoperability with other database systems

The Importance of Having a Plan

Before discussing the installation and maintenance aspects of file server database implementation, it seems fitting to bring up a topic that, though not directly related to database implementation, has a very direct and far-reaching impact on it.

If you are beginning to migrate a computerized database from large hardware platforms to a LAN-oriented system within your company, a database establishment plan is extremely important. A database establishment plan is a statement of all the tasks which must be completed to establish the database in the company. No one likes to talk about failed installations, but unless you cover the critical issues in database establishment, success can be extremely elusive.

Here are three suggestions to consider as you establish your plan:

- Get management's commitment to the database project—this ensures that you will have the support you need through the rough times.
- Select a small (but not insignificant) initial project to minimize the chances of your project being cut in a budget reduction.
- Choose people with a positive attitude toward LAN database migration to work on the project.

If you have the above items well in hand before database development begins in earnest, your chances of success will be significantly greater.

Installation Concerns

Although installation is one of the major topics of this AppNote, it is not really *about* installation in the sense you might expect. The points of installation that we will cover have little to do with "Insert the diskette in drive A." Our expectation is that you would prefer that. Besides, database vendors who have both market influence and developer ability usually produce high-quality installation routines. By high-quality, we mean that the installation routine is designed to get the program installed the way you want it, with a minimum of grief.

System Validation

Some installation routines even validate the system after installation. Such system validations are especially useful in complex database installations. Of course, this usually requires a knowledgeable programmer who can follow instructions and set up the right configurations.

Vendor Support

This brings up the matter of vendor support. Admittedly,

even the most astute installers are occasionally bewildered. We expect every serious database vendor to have a technical support department capable of guiding installers through those hard-to-solve problems.

Clients (end-users) who want to design and install their own database system are another matter, involving a separate set of considerations that we won't cover here. The information in this AppNote is tailored to LAN administrators who suddenly find themselves database administrators as well, and to database administrators looking to install file server based database applications and development packages.

Install Procedures

Installation of LAN-based databases really should occur in several phases. The first phase, obviously, is getting the database engine installed on a machine (possibly a standalone computer, to be replicated on the network). We have heard time and again from LAN supervisors and installers that the installation of a database is never really accomplished with the INSTALL program normally provided with the software. We can only concur, despite what the installation instructions say! In fact, it is only after the engine is installed that the fun of having and growing your own database system really begins, as you being to wrestle with maintenance issues such as backing up and upgrading the database files (see Figure 1).

Figure 1: Once you get the database installed, maintenance issues such as backup and upgrade become important.

Maintenance Concerns

Another of our interests is the ongoing system maintenance of a database after installation. There are very few database administrators who do not take an active role in the maintenance of the network database system.

What does network database maintenance entail? This is a question with a three-part answer:

- System maintenance
- Production maintenance
- Development maintenance

System Maintenance Issues

It seems that every database system has its own methodology for accessing data and different design considerations for implementing applications. This disparity may be due to the vendors' marketing departments trying to establish the uniqueness of their products. But what *you* have to go through to understand the installation and design techniques necessary to successfully maintain a network database can be something else entirely. It is almost always different than what you might expect after reading the advertising.

Because these differences are possible (if not inevitable) across a diverse LAN architecture, it is necessary to incorporate database management procedures with the day-to-day operation of the LAN. The LAN supervisor *must* work with the LAN database administrator, especially with regard to these critical issues:

- LAN administrator and programmer training
- System upgrade procedures
- Development of a backup plan
- Emergency response

Training LAN Supervisors and Programmers. One of the most significant aspects of installing a LAN database is ingraining the system's peculiar methodologies into your LAN management group and your programming teams. Often, these are two entirely different groups that have established different database (application) establishment procedures. Their methodologies for database deployment on the LAN often differ significantly; in many cases, their techniques are in direct opposition to each other and to the successful operation of the LAN database.

System Upgrade Considerations. Anyone who has ever had to re-install a database engine because of disk failure is undoubtedly familiar with the potentials for catastrophe (the restored version does not match the currently-developed constructs) or the loss of time and data due to irresolvable version conflicts.

There is also the consideration that almost every database vendor produces additions, corrections, and bug fixes to their system code. Eventually, these must be installed and tested. Because database system changes are usually distributed in the form of updates or upgrades, regular system upgrade maintenance can be both a concern to the database administrator and a performance booster well worth the time to install.

But for the LAN database administrator, changes in the database system represent a clear and present danger to the operational database system on the LAN. The danger lies in the chance that post-upgrade database systems may become nonoperational due to any number of causes (error in transmission, bug, version incompatibility, and so on).

Since most database administrators prefer a fully functional database system to early retirement, the prospect of software updates are usually viewed with at least some apprehension.

This leaves the LAN database administrator with only a few solutions with regard to upgrades:

- Never upgrade an established system
- Upgrade right away and let the production people worry about finding any bugs
- Test the updated software separately (install on a separate system and confirm that everything still works)
- Separate the development environment from the production environment and let development pre-test production upgrades

Of these options for administering database engine upgrades, we recommend the last two to minimize the worries of update/restore incompatibility.

Figure 2: Installing an upgrade to a separate system first lets you verify compatibility before upgrading the server.

Whatever model you choose for your database implementation— file server based, client-server, mainframe front-end server, peer-to-peer, or distributed—the database attributes must facilitate activities in recovery mode, as well as augment everyday work in system update and developmental code maintenance. For instance, the above example provides maximum protection of an operational system while an upgrade is installed and validated against current applications developed within that system.

Developing a Backup Plan. "Too many cooks spoil the broth," as the old saying goes, and too many people doing backups and restores can be disastrous to the database system once it is operational. Specific guidelines as to who backs up the database, what directories are involved, and when the backup is to be done must be established and understood by both the LAN supervisor and the LAN database administrator.

Emergency Response. Emergency response can be a concern where there is more than one person who can access a database backup. In case of disaster, it is much safer to have one source of authority for restoring the database, than to have two or three possible candidates each with their own procedure or each with the same procedure. Too many accidents can happen to destroy an application.

Update and Archive Issues

In a single-user database configuration, most of the database system maintenance issues are straightforward. Fortunately, many (such as update and archive issues) also apply directly to the multiuser networking model.

With only one person on a machine and having access to the database at a time, updates and upgrades can be applied directly to the local disk system after backups have been made and validated. In a multiuser system, verification and validation issues are more involved, as shown in Figure 3.

Figure 3: Before you archive your database system files, you need to verify and validate the installation, backup, and restore processes.

System verification determines whether the upgrade works correctly. The ease of system verification depends on what facilities the manufacturer has deemed necessary for checking it. If it fails, the original system can be restored rapidly from backups.

For the most part, there are rules for maintaining backups (for instance, the "tower" scheme, last in-first out, verify and validate). Restores, however, involve very little complexity because the order of events can be tightly controlled, one user at a time. Slightly higher levels of complexity might be encountered during developmental cycles in a single-user system, but those are due primarily to knowing which version of the application is currently in place and what changes have been made to the application since the developer's original modifications.

Development and Production Cycles

Once you begin to differentiate between developmental cycles and production cycles in a LAN database system, the complexity rises rapidly. Where in the single-user system only one person could be doing database work or database development, in the multiuser LAN environment development and production cycles can exist concurrently. This alone can make the backup exercise complex, even when the rules are clear.

Further, by incorporating a developmental- and production-based directory structure on the network (similar to the one shown below), different aspects of mission critical applications can be backed up at the appropriate time, without tying up needed network resources.

Since production files are likely to be in use most of the time, knowing which production files to update from developmental files and when to do so is another difficulty

in the multiuser database model. How do code and screen updates apply directly to data model integrity? Will there be irresolvable errors if production code and screens are updated while production data files are in use? These and many other questions arise in the maintenance of LAN database systems.

As a general guideline, you don't want to back up *open* production data directories (files) that may have transaction files in them. There is no way to know which changes have or have not been applied in the backup copy. Thus, no possible restore will be correct, unless the application is specifically designed to handle open transaction backups.

Production Maintenance Issues

Most database administrators who have worked with databases on the LAN face a nagging and recurrent question: Why does it take so many hours to run a production database, besides the time required just for setting one up? Issues such as how to update existing application code and system code, and how to devise data archive and restore procedures that really work in a production environment, can consume a great deal of time.

There are areas that need to be investigated by the LAN database administrator. One area not often mentioned in installation documentation is how to set NetWare file attributes (Read Write, Read Only, and so on) or which files should have which attributes.

From our experience, we find that attribute flagging has value in the areas of performance, security, and file integrity. In many instances, flagging the .EXE files as Shareable Read-Only (SRO) increases performance. It also lessens the chance of a LAN supervisor or equivalent accidentally modifying the existing .EXE files with an upgrade.

Development Maintenance Issues

LAN database systems today are changing at an unprecedented rate. With OEMs envisioning market growth from 15% to 50% annually, products are changing to meet requirements that would boggle mainframe developers. For instance, in the past year several major database vendors have introduced extensions to their products that allow them to use data from other databases.

Thus, another concern we will touch on briefly is the area of application development. While this is an area of particular interest to application developers, they often have difficulty in implementing database designs on LAN architectures unless they are familiar with the LAN as well as the database.

Often developers find themselves working on somebody else's LAN, where they are familiar with neither the LAN

supervisor nor the techniques used in the operation of the LAN. In these cases, and in instances where the LAN supervisor is ultimately responsible for the developed applications, it is extremely important for the developer to consider any additional training the LAN supervisor or LAN database administrator will need in order to preserve and maintain the application.

Another area of interest expressed to us is the subject of IS departments and programming staffs. We realize that most IS staffs are *not* looking to move their current DP systems to the LAN. Those that are seem more interested in client-server architectures (the subject of a future AppNote). But there are several reasons for looking first at file server based applications, before deciding upon client-server database platforms. One such reason is off-line program development for the mainframe.

Advanced Revelation from RTI

Having outlined a few of the issues in database maintenance, we can see how they apply to installing Revelation Technologies' Advanced Revelation.

Just as a note, all of our file server based database testing is being done on a Compaq 486/33 machine with 40MB of RAM and 11GB of disk storage. (Thanks again to Compaq Computers for their generosity.) To date, the server has been up, with the Advanced Revelation application running in one form or another, for six weeks. During that time, we have been loading and unloading NLMs for Btrieve applications and file/record creation statistics.

Over a period of two months, the server has been down only once, when we deliberately pulled the power to see how bad bad could be—but even that was not as catasrophic as we thought. To use RTI's terminology, "No runs, no drips, no errors" is a pretty good
track record. We look forward to passing along what we've learned in a future AppNote on large databases.

Advanced Revelation Installation

Among the many products that we have installed so far, Revelation Technologies, Inc.'s Advanced Revelation v2.00 and v2.1 for DOS/NetWare have proven to be the easiest in the default mode.

In fact, both the default mode and the alternative mode (install to a different directory) have shown themselves to work flawlessly in the DOS versions for NetWare, and acceptably in the OS/2 version.

In accordance with the considerations discussed above, the Advanced Revelation documentation recommends that the supplied system files be separated from the programming and data structures. In fact, the program effectively creates default directories for the storage of programmer created data constructs (files). During installation, Advanced Revelation automatically establishes a directory for data structures (\AREV\DATAVOL) and one for transactions (\AREV\TRANSACT).

Beyond the Default Installation

While this is a good start for anyone just starting with Advanced Revelation, we recommend that you create separate level one and level two subdirectories, as shown in Figure 4, for at least the programming and data files if there are several applications on the file server.

Figure 4: A suggested directory structure for separating production files from development files.

As shown in this figure, our file server based system uses \PROG and \DATA for the network level one subdirectories. We get the system to recognize them by implementing an ATTACH command specific to the particular LOGON (Advanced Revelation's, not NetWare's LOGIN) being utilized. Since all of this can be implemented similar to the AUTOEXEC.SYS file in NetWare or AUTOEXEC.BAT in DOS, the location of the program and data files becomes transparent to the client. (Developers can locate them instantly.)

Advanced Revelation Documentation

The earliest documentation efforts by RTI's predecessor (the company was purchased by one of its own product developers) received critical acclaim. The completely rewritten manuals shipping with the current product are also very well done and easy to read. The documentation consists of three manuals, each with its own charter to cover a specific group of functions. Each manual contains plenty of examples and usage suggestions.

However, the manuals are directed to the application developer, not the client. That is not to say that departmental developers and end-users with programming backgrounds would not benefit from this package, because they would. But *clients* should be given a manual on how to operate applications, not the manual on how to develop them.

Although Advanced Revelation's documentation for *network* installation (Chapter B1) is straightforward and mostly procedural, it is not extensive. My feeling is that database vendors feel that if they tell you what goes into maintaining their database, you are going to go off and buy someone else's product. Unfortunately, in many instances they're probably correct. But for that matter, where RTI's information on networks and network installation is not extensive, neither is anyone else's. The basic issues of network installation and maintenance are adequately described in the Advanced Revelation documentation. RTI has included everything needed to get the product installed, along with most of the standard caveats.

Additionally, RTI offers programmer technical bulletins as well as client, programmer, and developer support programs.

Optional Modules

Advanced Revelation offers several optional modules in addition to the basic developmental system. Right now, the

optional modules are aimed specifically at creating links with other data file structures. Currently, ASCII and dBASE file structures are provided free with Advanced Revelation. Several more (including Lotus, Btrieve, Sybase, and DB2) are also available.

Since RTI is one of the first companies to offer links to other application filing systems in the DOS/NetWare computing environment, we will discuss this feature in some detail.

Environmental Bonds

To use RTI's terminology, these links with other systems are "environmental bonds"—a concept which RTI has propagated far beyond the ordinary concept of links.

If you are familiar with linked spreadsheets, you might tend to think of environmental bonds as much the same thing. In some aspects they are the same, but only up to a point. The following examples might help clarify the difference.

In a single-user system, a linked cell in one spreadsheet updates cells in another spreadsheet because they are physically linked, as shown in Figure 5.

Figure 5: When you link cells in two different spreadsheets, changes in one are propagated to the other.

Similarly, if a database record is linked to a spreadsheet, the database record is modified whenever the spreadsheet is changed or modified.

The same basic theory applies to linked spreadsheets in a multiuser environment. When one spreadsheet is linked to a second spreadsheet, a changed entry in either spreadsheet updates the entry in the other spreadsheet.

But what happens if the second spreadsheet is in use by another user at the time of the modification? What happens when the first spreadsheet is filed? Is the second spreadsheet accessed at the disk or as a session on screen and updated? Are the spreadsheets interactive?

These questions and many others are answered by RTI's environmental bonds. In fact, with environmental bonds even more activity is possible, as shown in Figure 6.

Figure 6: With Advanced Revelation, you can establish environmental bonds with other database file structures.

Once you have established an environmental bond, the data /accessed with Advanced Revelation is actually stored in the alternate filing system's data file. Though this is nothing exceptional, how Advanced Revelation handles that data is

remarkable.

When an environmental bond is established with an alternate filing system, it is as if Advanced Revelation was running the application system that normally uses the alternate system as its native system. It updates indexes, establishes locks, and even notifies the end-user if that alternate application systems structure is inadvertently changed (by a developer) while the application is in use.

In short, RTI's solutions to providing an extensible application development tool, alternate filing system access, and database development look very good from here.

Installing Environmental Bonds

The actual installation of these extensible filing systems was a moment of truth that brought out two main points: the value of installing software on a local machine, and having separate production and development directory structures.

Installing on a Local Machine. When we were installing the RTI product by itself, installation was easy and performance (the speed of the install) was acceptable. Of course, for every rule there is an exception, and the exception proves the rule. We found one in the installation of third-party software.

Although possible on a network, the installation of the Btrieve Bond (from Icicle Software, Inc.) took a very long time. Justifiably, bonds are very complex objects and must by nature invade parts of the system that mere attachments would avoid.

When we began to install the optional modules (which is more of an upgrade), the installation performance dropped considerably. In fact, on one occasion we thought the process had terminated (during an attribute copy on the LAN). Later, after consulting with Pat McNearthney at Icicle Software and a bit of testing on our own, we found that the installation of the optional modules is best done on a local machine. Remember, the only reason to install the optional modules is to obtain access to file formats other than Advanced Revelation's native linear hash format.

For those reasons, it is probably faster to install the entire database system to a local machine, when possible, and copy the system to the network using network utilities. Since most of the product updates can be done on a local machine, this configuration provides several advantages:

- Backup of the original installation can be made an the local machine. A master copy of the program can be restored immediately if the original fails due to a media defect. Network traffic for installation is actually decreased, the installation functions happen locally.
- The original program can be updated from the backup without the network copy having to be brought back to the

local disk. Although if you combine the system files with developmental files this is not a good idea.

There are, of course, disadvantages, a major one being that if someone updates the network version without updating the local version, a disaster may ensue during the next update.
This procedure allows for the possibility that you might have damaged media, as well as improper installation and provides at least the basic backout procedure for a mis-installation or update. It also gives you an immediate idea about TSR and interrupt conflicts when tested on the local system.

Separate Directory Structures. The other point is that RTI provides a standard interface for product developers to interface with if they choose too. This interface is the update interface and provides a standard method by which product developers can send updates to clients.

When we tried installing all of the system at the local machine followed by copy to the network we found the process to be painless. RTI has progressively refined their installation and update procedure to a point where developers wishing to introduce third party products can utilize the installation and update process directly from the Advanced Revelation product.

When installing Advanced Revelation, it is definitely a good idea to install this product to a local workstation with the disk capacity needed. This provides the opportunity to perform the installation testing and reporting separate from the network, thus reducing the amount of I/O traffic to the server during the file copy and decompression process.

The way Advanced Revelation is designed, the basic files for screen design (TEMPLATES), menuing (MENUS), reporting (REPORTS), coding (BP files), and prototyping are initially provided in the \AREV (default) directory. But these structures are not cast in concrete; they can be copied or moved to other directories for the purposes of development.

We suggest that you make copies of the system structures for development to a \PROD (production) directory and \DEV (developmental) directory. In this way, application development and production code can be segregated from system structures and independently updated as needed.

Although this creates two or more copies of the original system files, it has proven well worth it for us. Yes, it's a bit redundant, but until the files in \DEV are populated, they do not take up an inordinate amount of room.

This type of structuring at the directory level also provides some facility for the system and developmental maintenance. System backups (of the Advanced Revelation database engine) that are made from the original installation (in the \AREV directory) can be archived until the

next upgrade/update is to be installed.

Things to Watch Out For with Networked Advanced Revelation

The following technical comments were gathered by Revelation Technologies' technical support and "Advantage" support groups. They alert you to several important considerations you should be aware of when running Advanced Revelation on Novell networks. For more information about these items and other problems you might encouter, contact Revelation Technologies.

"Disk Volume Full" Messages

Sometimes Advanced Revelation gives a "Disk Volume Full" error message when running on a Novell network. This might occur for several reasons:

- The current user has NetWare access limitations (such as a limit on the amount of disk space the user is allocated, or a network size limit on an Advanced Revelation subdirectory)
- The assigned owner of a file has been removed from the network (in either of the above cases, Advanced Revelation sees 0 bytes available on the disk)
- Ownership is lost when the original owner has been deleted or when the network is upgraded to v3.1x.

Possible fixes include the following: give the user full NetWare access rights to the directories and subdirectories in AREV that the user accesses; use Novell's FILER and SYSCON utilities to make sure the owner of the files is listed as a valid network user; copy all Advanced Revelation directories to new temporary directories, then copy them all back to the original \AREV directories (use either XCOPY in DOS or the filecopy function within Advanced Revelation); extend the disk size limit restrictions for the user and/or the AREV subdirectories.

Transaction Tracking (TTS) Issues

The Transaction Tracking System (TTS) feature in Advanced NetWare v2.1 and later has been known to cause Group Format Errors (GFEs) in Advanced Revelation linear hash files. To eliminate these GFEs, do not flag the Advanced Revelation files to use Novell's TTS option. Advanced Revelation has its own transaction tracking feature.

Of course, you can flag other non-AREV files on the same file server to use the TTS option. Just do not flag any files in \AREV or its subdirectories with NetWare's Transactional attribute.

Changing Drive Mappings

Another problem occurs when users change their drive mappings while temporarily suspended out of Advanced Revelation. Advanced Revelation initializes and sets up its locking scheme table only when a user first logs in to AREV.

For example, if a user suspends out of AREV, attaches to another Novell file server, then comes back into AREV using the "EXIT" command, no record or file locking will occur for any data on the newly attached server.

The solution is to log off of AREV (using the Advanced Revelation logoff, not Novell's LOGOUT command), attach to the desired Novell file server, then log back in to AREV. If you need access to multiple Novell file servers simultaneously, set up appropriate AREV volume names for the different Novell servers and attach to them automatically via the AREV "LOGON" record in Advanced Revelation's VOC (vocabulary) file. The locking scheme will then work correctly when logging into Advanced Revelation for all attached Novell file servers.

Network Hardware Considerations

Hardware considerations also affect Advanced Revelation and Novell networks. Poorly placed network cabling (such as unshielded cables behind refrigerators, over fluorescent lighting, or around strong magnetic fields) can mysteriously trash or corrupt data, as manifested through GFEs in Advanced Revelation. Also, older network interface boards in a workstation or file server sometimes have corrosion on their connector edges, which can also cause GFEs in Advanced Revelation.

Solutions include using shielded network cabling and selecting a network cable route away from interfering magnetic or electronic signals. You should periodically pull out network interface boards and disk controller boards, clean the connector edges (scrubbing the edge with a clean pencil eraser works well—take care not to get eraser crumbs inside the computer), and reseat the boards firmly in the expansion slot.