

Backup and Recovery Strategies - Part 1

Backup and Recovery Strategies for Mission-Critical 4D Server Databases: General Principles of Database Backup and Recovery

By Walt Nelson, Manager, ACI Technical Support

4th Dimension Technical Note 97-1

Technical Notes for 97-01-02-January/February 1997

Introduction

The trend that we are seeing in ACI Technical Support is that thousands of customers are using 4D Server for mission-critical data storage. Daily, we are talking with customers who have 4D Server installations with as many as 100 users connected, and with data files as large as five gigabytes. The customers view these applications as "mission-critical," meaning that the organization cannot function properly if the system is down. Often, these databases are running 24/7: twenty-four hours a day, seven days a week.

You can depend on 4D Server to give you mission-critical, 24/7 stability, **if** you apply some common sense to your Database Administration duties and **if** you follow a few simple rules of backup and recovery. Many customers have been able to maintain 4D Server installations that run for months—seven years—with little or no downtime. The purpose of this two-part technical note is to provide you with the information that you need in order to set up and maintain an effective backup and recovery strategy for a mission-critical 4D Server application.

This technical note is in two parts:

Part 1 - General Principles of Database Backup and Recovery

Part 2 - Automatic Backup and Mirroring with 4D Backup. This part will be presented next month.

General Principles of Database Backup and Recovery

In the Information Systems (IS) world, the person responsible for database backup and recovery is called the Database Administrator (DBA). In Part 1 of this technical note, we will discuss the duties and responsibilities of the DBA with regard to backup and recovery of a mission-critical 4D database.

What is a mission-critical database?

As we stated earlier, to say that a database is mission-critical means that the organization cannot function properly if the system is down. This means that the information in the database must be accurate, it must be timely, and it must be available during the hours of operation of the organization. In many cases, keeping a mission-critical database running is vital to the success of the organization.

As you will quickly discover when you become a DBA, maintaining a mission-critical database is a very demanding responsibility. When the database is down or if the data gets corrupted, you have a potential disaster in situations like:

¥ Invoices don't get issued

¥ Statements don't go out~Nor even worse, they're wrong

¥ Payroll checks are wrong~Nor even worse, they don't go out!

The first time that anything like this happens to you, you will really begin to understand the implications of the term "mission-critical."

In this technical note, we will discuss the following three simple rules of backup and recovery:

¥ Set your stability goal

¥ Make backups, backups, backups

¥ Take standard precautions

Rule #1: Set your stability goal

Your first step in creating a backup and recovery plan is to set your stability goal: define exactly what the term "stable environment" means to you. Like all goals, your stability goal should be clear, simple, and measurable. For a mission-critical database, your stability goal might be:

"Never down for more than 15 minutes."

This means that, during your organization's normal working hours, your database should **never, ever** be down for a period longer than 15 minutes. Having a goal is very important; you need a standard to live up to. You will recall that, in the Introduction, we said that a database is "mission-critical" if the business cannot function properly when the system is down. When the system is down, the business is down. As the Database Administrator of this mission-critical database, your job is to make sure that the system never goes down~Nor if it does go down, it is only down for a few minutes.

Rule #2: Make backups, backups, backups

Many people believe that computers are supposed to be perfect; only humans make errors. Unfortunately, that is not so; computers are not perfect. Computers may not make errors as often as people do, but computers do sometimes make errors. Let's look at some interesting numbers:

Although most 4th Dimension data files start out small, it is not unusual to see data files that have grown to 100, 250, 500, or even 1,000 megabytes (one Gigabyte) in size. Suppose, for a moment, that your data file grows to 400 megabytes. That means that you have four hundred million pieces of information stored in the file~quite a record-keeping task. If your hard disk makes a single mistake and loses track of even one of those 400,000,000 pieces of information, you will have Data Corruption.

Because you may need to call on any one of those 400,000,000 pieces of information at any time, 4D has to know, at all times, exactly where every piece of information is located, and exactly how each piece relates to the other 399,999,999. Each time you create a new record, 4D must decide where to put the new record and how to keep track of it. Each time you change a piece of information, 4D has to find the old one, delete it, and save the new. And, if the new information takes up more space than the old, 4D has to find a new place to store it. This data-shuffling exercise takes place thousands of times a day in a heavily-used 4D database.

In the process of all this data-shuffling, if the network, the CPU, or the hard drive fails, data corruption can result. The wonder is not that data corruption happens; the wonder is that it happens so seldom!

Sooner or later, every electrical or mechanical device will fail. The question is not whether something will fail, but when it will fail. When it does fail, you need to have a backup that will get you up and running again quickly. If your job is to keep the system running no matter what happens, then you must accept the fact that every component can fail. Your job is to have a backup system in place, ready to go on-line immediately, when the failure occurs. The key to eliminating downtime, then, is backups, backups, backups.

The principle of backups

Let's look at an example of backups from real life. If you want to call a business associate on the phone, you must get the phone number from a directory. That directory might be stored in your head, but it is still a directory. If you have forgotten the number, you can't call your associate. You know that the business is still there, and you know that the business has a phone, but you can't call because you don't know the number!

But I can still call, you say. I have my Rolodex, I have the Phone Book, and I can even call Directory Assistance. To which we say, Exactly! Your Rolodex, the Phone Book, and the telephone company's Directory Assistance service are the three backups to your mental directory of your business associate's phone number. This system is foolproof; you will always be able to find out your associate's business phone number from at least one of the three backup sources.

If we want to classify those three backups in a way that means something to us in the computer world, we would say that you had an on-site snapshot backup (the Rolodex), an on-site full backup (the Phone Book), and an off-site full backup (Directory Assistance). And so we can state a Principle of Backups—keep three levels of database backups:

¥ On-site snapshot backups, stored on other drives within your office

¥ On-site full backups, stored on other drives within your office

¥ Off-site full backups, stored outside your office

Managing your incremental backups

Incremental backups are backups that reflect the status of the data at specified points in time. The principle of incremental backups is not new; professional MIS departments have used the concept of scheduled incremental backups for years. They usually set up the following schedule for backing up mission-critical data:

¥ Daily incremental backups, keep them for a week

¥ Weekly incremental backups, keep them for a month

¥ Monthly incremental backups, keep them for a year

The following are the specific steps that you can follow in making your incremental backups. Next month, in Part Two of this technical note, we will give you several hints and tips for setting up 4D Backup to automate the process.

For your Daily backups, create five disks labeled Monday, Tuesday, Wednesday, Thursday, and Friday. On Monday, at the end of the day, after you have done your end-of-day tasks and printed the daily reports, insert the Monday cartridge so that 4D Backup can make the scheduled Monday backup. Then do the same on Tuesday, Wednesday, Thursday, and Friday. If your hours of operation include Saturday or Sunday, you should add those days to the backup schedule.

For your Weekly backups, create five disks labeled Week 1, Week 2, Week 3, Week 4, Week 5.

On Friday, at the end of the day, after you have done your end-of-week tasks and printed the weekly reports, use the appropriate week's cartridge to make a new backup for that week number.

For your Monthly backups, create twelve cartridges labeled January, February, March, and so on through December. At the end of January, after you have done the end-of-month posting and you have printed the month's hard copy reports, use the January cartridge to make the new January backup. Do the same at the end of February, March, and so on through December. Your December backup should also serve as the full end-of-year backup.

Off-site storage of backups

One philosophy of backup management is to "hope for the best, but plan for the worst." Perhaps the worst that could happen to you as a DBA would be a fire or a natural disaster that completely destroyed your office, all of your equipment, and all of your computer files. If that happened and you had backups that were stored off-site, you could purchase new computers, move to a different location, and be fully operational within a few days. But if you didn't have backups and you had to input all that data again, you still might not be fully operational several months after the disaster!

Here are some rules for storing off-site backups:

¥ At least once a week, make an extra copy of one of your daily incremental backups and store it off-site.

¥ Each month, store copies of your monthly incremental backups off-site.

¥ Make sure you have off-site access to a computer and a disk drive that can read your off-site backups.

Making self-contained backups

Another important principle of backups is that they must be self-contained. By this we mean that you should assume, when making a backup, that your entire office has been destroyed—including all file servers, workstations, and disk drives. If that happened, what would you need, along with a new computer, in order to get back in business? Obviously, the answer is that you would need every file and every utility that it takes to run your database: 4D Server, 4D Client, 4D Tools, Structure, Data, 4D Calc/4D Draw/4D Write Formats, Report formats, Search formats, 4D Extensions, disk utilities, and of course, your system software.

Keep in mind that this self-contained aspect is even more important for monthly incremental backups, which are historical. A year from now, when you need to open that data file, your current structure, system, and version of 4D or 4D Server may all be different from the versions that were current at the time that you made the backup. To ensure 100% compatibility, you should open the historical structure and data file with the same version of 4D and the System that you were using when you created the backup—in short, you should backup everything that you would need in order to turn on the computer and run your application.

Rule #3: Take standard precautions

Over the years, IS departments running 4D Server applications have developed some standard precautions that they take in order to minimize the risk of downtime or data corruption.

Use a dedicated server

A non-dedicated server is one that is doing double-duty as a user's workstation, and at the same time as the server for the database. If you are running 4D Server on a non-dedicated server, you are far more likely to experience data corruption. For example: if the combination workstation/server happens to have a system crash (on Windows, a General Protection Fault or Exception Error) while running some other program, the crash could freeze the entire machine, corrupt the memory, and bring down the 4D database as well. If the database happened to be saving a record or updating an index at the time of the crash, that record or index will probably be corrupted.

The only setup that we can recommend for a multi-user 4D database is 4D Server running on a machine that is dedicated to 4D Server. You may think that you cannot afford a dedicated server, but when you see the higher incidence of data corruption that occurs with non-dedicated servers, you may come to the conclusion that you can't afford not to have a dedicated server.

Install a UPS system or a line conditioner

A UPS (Uninterruptable Power Supply) activates in a split-second whenever it senses a significant drop in power. Most UPS systems are intended to let you run your computer long enough to shut down in an orderly fashion, thus avoiding data corruption. The ideal setup is to attach a UPS to your server as well as to all workstations. If that setup is too expensive, then you should attach a UPS to your server and to the most active workstations. UPS units have different time-operation capabilities. Fifteen minutes are usually sufficient to allow for an orderly shutdown.

A Line Conditioner is a less expensive alternative that provides protection against the peaks and valleys of commercial power sources. The line conditioner does not provide backup power, but it does stabilize your power at a constant voltage. If you cannot afford a UPS, at least make sure you put the server on a line conditioner. A line conditioner costs about 1/4 the price of a UPS unit. A UPS unit also serves as a Line Conditioner, so you don't need both.

Thoroughly test new releases before installing on your server

As a DBA responsible for a mission-critical application, you should always lean toward the conservative side when it comes to new releases of hardware and software. Before installing new machines or products in a real-time production environment, you should always thoroughly test them on a spare copy of your database. You should maintain enough space on your server and clients so that you can set up the test in your actual environment, with copies of your structure file, data file, and of 4D Client that are clearly labeled as test copies. Then, ask some of the users to help you by sitting down and entering real data; running real reports; and doing real searches. Compare the results to what you get with the production system. Only after you are satisfied with the results of these tests, should you install the new release in your production system.

Use the WEDD Resource.

This WEDD resource enables you to match a 4D structure file with a specific data file, thus preventing a user from mistakenly opening a data file with the wrong structure. In the latest version of the structure, the developer may have added new files, fields, subfiles, indexes, and so forth. If 4D opens the database with this updated structure, 4D immediately senses that there is a change and it re-organizes the data and indexes to accommodate the new structure. Later, if a user launches that data file with an old structure that does not have the new files, fields, and indexes, 4D detects the differences and attempts to reorganize the data again to accommodate this new structure. The probability of this type of user mistake, resulting in data corruption, is significant enough to cause concern.

Using the WEDD resource, you can make sure this type of data corruption does not occur. The WEDD resource is merely a password. By storing an identical password in the structure file's WEDD resource and the data file's WEDD resource, you indicate that the structure and the data are married (thus the name WEDD resource). Upon startup, before it tries to interpret the data file, 4D checks the data file's WEDD resource to see if the password is identical to the password in the structure file's WEDD resource. If they match, that is fine; if not, 4D notifies you that this structure file and this data file do not match, and it does not attempt to launch the data file.

Run 4D Tools (Check and Fix option) At Least Once a Month

An ounce of prevention is worth a pound of cure. As a DBA, you can save yourself a great deal of trouble by following that wise advice. At least once a month, even if you have no reason to suspect problems, you should run 4D Tools (Check and Fix option) on your data file. By doing so, you can detect and fix minor problems before they become major.

Summary

You can depend on 4D Server to give you mission-critical, 24/7 stability, if you apply some common sense to your Database Administration duties and if you follow a few simple rules of backup and recovery. Many customers have been able to maintain 4D Server installations that run for months—Even years—with little or no downtime. In Part 1 of this technical note, we have presented some general principles that you should follow in planning your database backup and recovery strategy. In Part 2, we will give you some tips and tricks for implementing Automatic Backup and Mirroring with 4D Backup.

See Also

[Backup and Recovery Strategies - Part 2.](#)