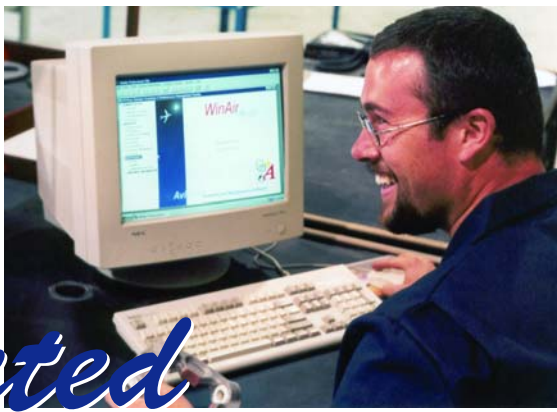


As seen in the  
February March 2005 Issue of AMU



GETTING CONNECTED  
by  
GAVIN MILLER

# GETTING *Connected*



STORY BY GAVIN MILLER,  
MPLAN SOLUTIONS, INC.  
GRAPHICS BY JASON NORRIS

**AN INTRODUCTION TO THE VARIOUS WAYS COMPUTERS CAN BE EMPLOYED IN THE MAINTENANCE DEPARTMENT.**

*It* should not come as a surprise that computers can be used to make your maintenance tracking and inventory management tasks easier, but the number of ways an electronic system can be configured may surprise you. If you ever need more than one computer sharing the same data, there will be decisions to make regarding how they should be connected—now and in the future. You need to choose appropriate software and hardware that will work together to solve problems, not create them, and will be able to grow with the business. Like painting the hangar floor, choosing the wrong place to start can leave you standing in the proverbial corner wishing you had employed a little more forethought. Whether you are considering the purchase of the first computer for the maintenance department or planning to implement a new Wide Area Network solution, a reasonable understanding of the alternatives will allow you to choose a method—and a software provider—that meets your current objectives and can grow with you.

A simple system might start by providing scheduled maintenance forecasting on a single PC in the office. Then another computer for the stores department might need to be added. If the software you are using is capable of doing both in an integrated manner, then the two computers will need to be sharing the same data. The operations department might have a flight planning and crew scheduling system which can be integrated with maintenance planning, and the accounting system may want to interface with inventory control. And it doesn't stop there—today's software and

technology allows you to implement a cost effective inventory, purchasing and maintenance management system that can be used by a line mechanic on the hanger floor with a wireless PDA, while the maintenance manager with a laptop plugs in information during the pre-purchase inspection on an aircraft on the other side of the world. Obviously, choosing the right software providers are a big part of such a system but so is proper planning of the technical structure.



Simple Network

In the past, the cost and complexity of computerized tracking systems made third-party solutions popular—systems which were designed and managed by outside service providers or the aircraft manufacturer. The user would supply information regarding the utilization of their aircraft and the maintenance which had been completed (usually on paper), and the service provider would plug this information into their system and tell the user which maintenance would soon be due (again, usually on paper). This is still popular in the business aviation community. But in today's regulatory environment, many operators must

design an appropriate maintenance schedule for specific aircraft types in their particular operating environment and then have it approved. This does not lend itself well to leaving the control of the tracking system in the hands of someone else with a cookie-cutter system and as a result many operators are taking matters into their own hands. With the advent of Windows-based software and network solutions for the masses, a sophisticated, easy-to-use multi-user maintenance tracking system which integrates with inventory, purchasing, production and billing can be implemented for less than the cost of a good maintenance clerk—often for less than the cost of a single maintenance planning contract with a service provider.

## THE SINGLE COMPUTER

Many a maintenance manager uses a spreadsheet to schedule their maintenance. Anyone with a little computer sense and a reasonably capable work station can build a simple list of recurring maintenance tasks and enter formulas that will help them to predict when events will come due. This is actually little improvement over a paper-based system and there are several disadvantages to it. A careless user can easily alter the formulas or figures in the spreadsheet and the mistake may not show up for a long time. There is no history record for any of the maintenance performed—when has it been done in the past, who did it, etc. But perhaps the biggest problem is that a single user works with a single “file” in the computer. If two people need to access or update the information, things get complicated.

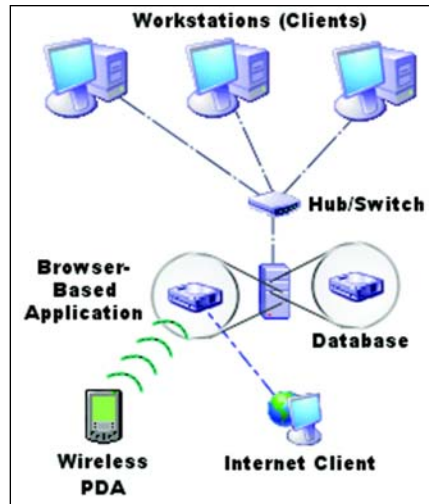
## DATABASES

The first step down the road to a more capable system is an application (software program) which uses some form of database to store information. A database uses “tables” of information (rows and columns) which are similar to spreadsheets but usually there are many tables containing related data which is managed by some form of “user interface” so that the software designer can build in controls which limit what the user can do. Your email program is probably a simple example of a database application. More capable database programs usually incorporate the ability for multiple users to access the data at the same time. The better ones also provide controls over what type of access each user will have so that a security system can be defined. The fact that more than one user can access the data at the same time introduces the first technical difficulty—how to connect more than one computer to the database.

## PEER-TO-PEER

Peer-to-peer is one of the simplest network forms. Most modern Windows-based computers have the necessary hardware and software to allow the user to connect two computers together with a network cable (a special “crossover cable” is used to connect two computers directly) and share resources including printers, scanners, data and even software. A good multi-user application will incorporate controls so that two computers running the software can share one database stored on the main computer, commonly called the “Host.”

When the time comes to add a third workstation, another device is required—a network hub. A hub is basically an inexpensive box into which you can plug multiple computers using standard network cables so that they can all share each other’s resources. This way, several computers can share the data which is stored on the host. But problems begin to develop including performance and stability. Since the host is actually a functioning workstation, it has to perform all the functions required by the person



External Connection Options

using it, and supply “read and write” services to the other workstations accessing the database. In addition, any problems experienced on the host computer will affect all the users of the application (if the host computer “locks up,” it can result in poor data integrity or even corruption). To mitigate these kinds of problems, the host computer can be relegated to the job of hosting the database and not be used as a workstation but the next step in the growth of the system is the move to a Client-Server network architecture.

## CLIENT-SERVER

In a client-server system, shared data and files are stored on a dedicated computer called a file server. The workstations to which it “serves” data and files are clients. A special operating system is installed (such as Microsoft Windows Server Edition) so that it can also provide other network services such as controlling user access to network resources, or the Internet, or providing data backup services. For a simple low-end database application, this is the typical architecture. Although performance is improved over a simple peer-to-peer network, high use of a multi-user database application will begin to reveal other limitations.

Performance will suffer as the amount of use, the number of users and the quantity of data increases. This is because the server is really just a “dumb” computer that stores the files required by the clients. Data tables such as your inventory listing can

become very large files. If your application is a good one, it will store a lot of transaction history and these files will grow larger the longer you use the application. Each time a client needs to access data from these files, it must retrieve them from the server. The bigger the files are, and the more network traffic there is, the more the performance of the application will suffer. This can be improved by increasing the communication speed over the network, by using switches instead of hubs to reduce needless network traffic, and by improving the capability of the server but these are really just patches to a system which is no longer adequate.

Another problem can begin to surface as the performance of the network is reduced. Since the file server is simply a storage place for data files, the



Peer-to-Peer

software running on the clients is in control of what changes are made to what files. If the software on one client is in the process of making a complex series of related changes (e.g. receiving parts, thereby updating stock on hand, backorders on a PO and inventory history files all at once) and the user is working faster than the computer can respond, improper updates can result. Good software will have checks to ensure that two clients can’t make simultaneous changes which cause problems but this safety can be compromised if network response becomes too slow. In effect, users can partially or completely overwrite each other’s updates and cause serious problems. The best database systems address all of these issues by using a database server instead of a simple file server.

## DATABASE SERVER

A database server is actually a piece of “intelligent” software which runs on the server computer and takes charge of the data. Microsoft SQL Server and Oracle are popular examples. Although client computers still share the data, the database server understands the relationship between the tables in the database and does not allow data corruption such as that described above. This is because the clients don’t actually make changes to individual data files directly—in effect, they tell the server what changes to make and it ensures that they are made properly and completely. In a well-designed database application which uses this type of architecture, the server knows what tables need to be updated to perform a complete “transaction.” In the example above, the server would make the required changes to the Inventory, PO and History data tables when a part is received on the client computer. If all of the required changes cannot be performed properly for any reason, the entire transaction does not happen. This is referred to as a Relational Database Management System (RDBMS) in contrast to “flat” data files.

There are several benefits to this type of system design. Since the management of the data is largely performed by the server, much less information needs to flow over the network and therefore performance is vastly improved. If the application is well designed, the client computers will only be sent the minimum amount of information that the user needs to see at once. Data integrity and security is much better because the server is in control of the changes that are made.

Another big advantage to this type of architecture is that connections to the server are useable even if they are relatively slow because much less data has to move between the server and the client. This means that internet or wireless connections to the server from remote locations is possible (e.g. from other bases or from aircraft in the field). A flexible system might even



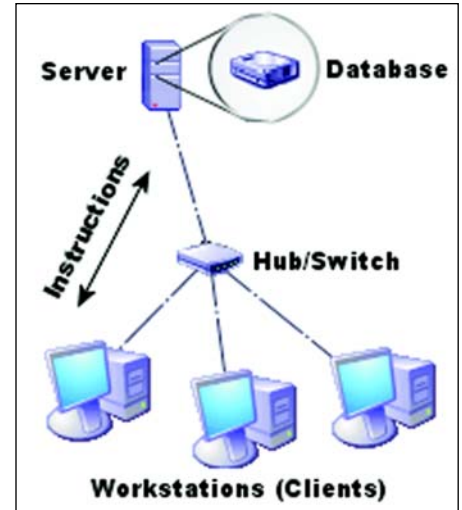
Client-Server

allow access through interface applications which reside on the server so all that is required on the client computer is an internet browser.

A common example of this type of system is banking over the internet. When you are sitting across the desk from your loans officer, she is accessing the same data in the same database but her computer has a different application on it which let’s her do more than you can from home using the software supplied by the bank’s server. Similarly, your maintenance management system can be accessible to the purchaser on a desktop computer in the stores department, a mechanic on the shop floor using a handheld wireless device or a manager with a laptop and a cell phone attending a trade show.

## WHAT IS RIGHT FOR YOU?

If you are shopping for software, make sure you understand on which of the above types of computer system it can be installed. If your organization is of sufficient size that you already have a client-server network and will need several workstations, you should probably be looking at software that can make use of a proper RDBMS such as SQL Server. If the software can only use flat data files in a simple Peer-to-Peer or Client-Server network architecture, be prepared to be dissatisfied by the performance once the size of your data files begins to grow. Unless you never expect to have more than a few part-time users, ensure that the software you choose can grow with you without the need to re-enter data when your needs



Database Server

change. Make sure the system you choose is well supported and has a strong history of keeping pace with the needs of its users and the capabilities of the computers on which it runs. Setting up a new system is enough work without the need to do it all over again in a few years!

Here are some good questions to ask the developer: Can multiple users perform the same tasks at the same time? Is there a security system to control which users have access to which functions? Does it use simple database files or a proper database server? Can it start small and simple and move up as your needs increase? Can it be accessed using an internet browser from any computer? ■

*Gavin spent 20 years fixing helicopters (and the occasional round-engine de Havilland) but in 1999 he went to work for Bert Vergeer and his team at AV-Base Systems Inc. of London, Ontario. Bert is the principal designer of WinAir, now the leading aviation Maintenance and Inventory software in Canada. With a similar background — fixed wing (Bert) and helicopters (Gavin)— they combined maintenance experience and helped to make WinAir what it is today. Now Gavin spends most of his time running MPLAN Solutions Inc. which includes building maintenance schedules for WinAir and travelling around the world training maintenance departments on their new software.*

*For more information about maintenance tracking systems, contact Gavin Miller at MPLAN Solutions Inc. [www.mplan.ca](http://www.mplan.ca).*