doors & hardware

AUGUST 2011

ADVANCING LIFE SAFETY & SECURITY SOLUTIONS

SECURITY

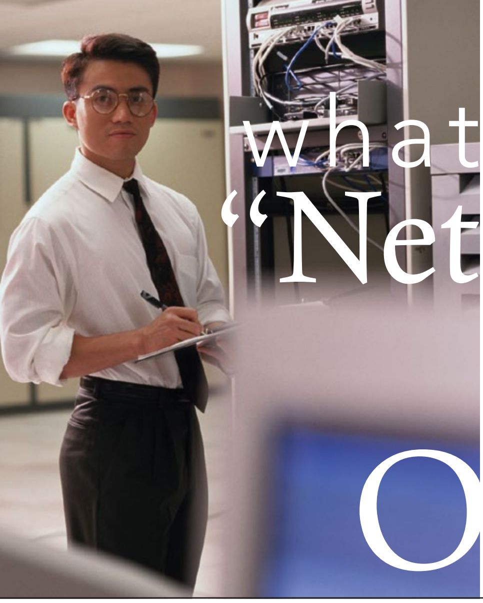
THIS MONTH

D&H talks to industry members about security trends in their region. We also examine egress width requirements, decode auto operators, and find ways for distributors to improve payroll control.

INDUSTRY **In the Know**

Advance the FDAI initiative before it's formally adopted, and get a jump on your competition! ***See page 38***

A PUBLICATION OF THE DOOR AND HARDWARE INSTITUTE



what makes a "Network Application"?

By Paul Kirsch

One of the most interesting aspects of running a support department is observing trends in customer inquiries and identifying common points of confusion.

By far the most common question we hear in both the sales and support departments is: "Is this a network application?" The question seems simple enough. "Yes" or "No"; "It is" or "It isn't." Simple, right? Not so much.

To a software developer, the terms *network application* or *network capable* really mean nothing at all. It's equivalent to asking if your new car is "road capable." There are so many different ways that a piece of software can be installed on a network, and even more ways that its resources can be distributed on said network, that a "yes" or "no" answer to this question tells you absolutely nothing.

In most cases, if a salesperson does answer you, he is either guessing at what you're referring to by your inquiry or more likely just giving you the answer that he believes you want to hear. Quite honestly, virtually every piece of software is a "network application" or is "network capable" to some degree. It just depends on how you look at it.

If we were to attempt to come to a meaningful answer to this age-old matter of network compatibility, we would be best to begin by looking at the various components of both the network and software in question.

The Network

While there are as many different network configurations as there are stars in the sky, all of them can be broken down into three major groups:

Peer-to-Peer Networks: This type of network is most commonly found in homes or extremely small offices with only a few users. It consists of several computers, networked together, with no designated server or main computer. Based on security settings, each PC can optionally share its files and access files from all the others. There are also usually printers and an Internet connection that is being shared by all of the machines. What defines this type of network, however, is that no PC is more important than any other. Each PC is intended to be used by a person, and there is no established central server on which the other PCs are dependent. If any machine goes down, the others are unaffected.

Server-Based Networks: This type of network is most common in commercial settings and is defined by one or more common servers that are accessed by some or all of the PCs connected to the network. Often the PCs within a server-based network also have peer-to-peer functionality, in that they can access each other's files, but the primary intent is to have shared resources stored on servers. This is essential to maintaining optimum performance. Unlike peer-to-peer file-sharing, servers dedicate 100% of their resources to "serving" files to the various users at top speed. The important thing to remember is that when the servers go down, business comes to a screeching halt.

Thin-Client Networks: This is by far the least common and most radical configuration you're likely to find. In this environment, the only computers on the network are the servers themselves. They don't just serve files, however; they also serve processor time. There are no PCs needed on the users' desktops, just "thin clients" (otherwise referred to as *appliances* or *dumb terminals* because they contain no software or operating system). All they can do is connect to the server and allow you to control software that is running on them. The servers "slice" up their processor resources and run the software application on behalf of each user. Although extremely simple to install and maintain, the drawbacks to this configuration are huge. Performance typically suffers because users are ultimately sharing a processor. In addition, those thin clients are completely useless when the server goes down.

The Software

Although this can vary between software packages, most can divide their resources into three major categories:

Software: The first major component is the actual software. This is usually located in the "Program Files" or a similar folder and contains the program itself.

Static Data (i.e., Libraries): Often thought of as part of the software installation, this is the data component of the software program. This data is never changed by the users. In the case of architectural industry software, this would include things such as location libraries and manufacturer catalogs.

Dynamic Data (i.e., User Files): The final component is the files that the users create. Their individual data files and the files that are shared among multiple people in the company all fall into the general category of "dynamic data."

Where Does It All Live?

When installing software in a network environment, the big question is where to locate each of these three major groups of files. Any one or all of them can be located either on the server or on any of the individual PCs. As long as any one of them resides on any machine that is accessible by others on the network, you have yourself a "network application." Starting to see why this term is so ambiguous?

The next bit of news is really no more encouraging. There is simply no correct answer as to where these components *should* be situated. This is entirely a matter of opinion. For each choice, there are pros and cons. Every software developer will make decisions based on the way he configures his software based on his own priority structure. Trying to examine every conceivable combination and option is beyond the scope of this article—people have written entire books on the subject. But we will look at some of the basics. The first question is where to install the software applications themselves: on the server or on the individual desktops. Each one has its advantages, and it's up to users to decide what is most important for them.

In terms of functionality, it's a complete toss-up. In either scenario, data can be freely shared between people and stored in either a central repository or on users' individual hard drives. This type of organizational decision is a business matter and is not predetermined by software residency.

With respect to said residency, there are of course a few considerations. An argument can be made for both strategies, but there is a clear advantage to one over the other.

1. **Ease of Setup and Maintenance**
   In this category, server-based installations are the clear winner. With the applications installed centrally, there is only one machine to install and only one to update moving forward. For this reason, this has always been the "hands down" choice of IT mangers and network administrators. After all, it certainly makes their lives a lot easier. All of this convenience, however, comes at a price.
2. **Fault Tolerance**
   Without a doubt, the most important consideration in designing an application environment for a business is the notion of fault tolerance. Simply put, this concept speaks to the amount of component failure that the system can endure while still allowing work to continue. When software is installed on the server, every user is completely dependent on that machine functioning. If the server or any part of the network connecting the users to it goes down, everybody gets a holiday. Rule one in any software installation should be "Business is never interrupted." When applications are installed on each individual machine, one could literally rip out the network cable from the back of it and business could go on. Users could still access the software and work locally. In cases of a network failure, data files could still be shared peer-to-peer and copied back onto the network drives when they are restored.
3. **Flexible Updating**
   By installing software individually, users have the flexibility to install updates on their own schedule. New versions of a program can be evaluated by select users before the entire company is committed to the change. In addition, new software can be gradually phased in, allowing for the progressive training of company staff. With a centralized installation, everyone adopts the change at the same moment.
4. **Mobile Computing**
   Distributed application installation allows for the mobile devices such as laptops and remote PCs that need only connect to the network to update the user data files. A home-based user is out of luck if his Internet connection goes down, and a laptop computer is completely useless on an airplane if someone is dependent on connecting to a file server to run his software.

At the end of day, the choice is up to the buyer. The important thing to take way from all of this is the fact that all network applications are not created alike, and despite the obvious lure of a "centralized" solution, there are many other important considerations that should be made.

About the Author: *Paul Kirsch is the President and CEO of AVAware Technologies. He has been involved in designing and writing integrated software systems for nearly 25 years. Paul can be reached via email at pkirsch@avaware.com or via the AVAware website at www.avaware.com.*