



Developing with 4D Server Part 1: Advantages of 4D Server Multi-User Development

By: Walt Nelson

Introduction

In July, we discussed the concept of using two of 4D's powerful built-in features, subfiles and relational lookups, to improve performance. In August, we continued our review of 4D's built-in features, describing seven powerful uses of 4D Choice Lists. This month, we are continuing this review with a discussion of the advantages of using 4D Server/4D Client for multi-user development.

This review is comprised of three parts:

- Part 1 - Advantages of 4D Server Development
- Part 2 - Setting Up a Multi-Developer 4D Password System
- Part 3 - Naming Conventions for Multi-Developer 4D Projects

Let's continue with Part 1 of our discussion: Advantages of 4D Server Development

Problems that 4D Server development can solve

Have you ever experienced any of these problems?

- You want to work on a database at home and discover that you left the latest version of the structure at the office.
- It has been several days since you worked on this particular project. You are at home, you want to work on it now; but you can't recall if the version at home or the version at the office is the latest version.
- You are working on a multi-platform application (for example, Mac and Windows). You create some layouts on your Mac and they're perfect; however, when you run the application on Windows, you don't like the look of the fonts.
- You are working jointly on a project with two other developers. Merging the code of three people is a constant headache: having coordination meetings, resolving naming conflicts, correcting errors that result from one developer's objects overriding another developer's objects.





4th Dimension

Technical Note 96-38

- You are writing user documentation for an application. You create some screen shots on Windows, they look fine; you finish the entire documentation and you ship the application. However, when your Macintosh customers open the documents on the Mac, many of the screen shots look vastly different than they did on Windows: some screen shots look pale, others look as if they are in reverse video.

If you have ever experienced any of the above problems, perhaps you should consider doing your development in 4D Server, rather than in single user 4th Dimension.

To get a balanced view of the subject, let's discuss the pros and cons of doing your development with 4D Client while connected to 4D Server. First the disadvantages.

Disadvantages of 4D Server development

Equipment

The first obvious disadvantage of 4D Server development is that you will need a server machine and a certain number of user licenses. However, the server does not have to be a very fast machine—you can even run the development copy as a second instance of 4D Server on an existing machine. In this case, the only additional equipment expense would be for additional RAM, so that you do not take RAM away from your primary application. There will be a slight performance penalty; however, as a developer, you will not be taxing the resources of the machine very much. Most of your requests will be for procedures and layouts, which are small compared to data requests by users.

Speed over remote connections

The biggest single disadvantage of developing with 4D Client while connected to 4D Server is reduced speed: it takes about 10 times longer to sign on to the database over remote connections than it does locally over Ethernet. Once you are connected, however, speed is not much of an issue. After you copy the object (a procedure, layout or script), to your local machine, then you are working at the speed of your local machine until you save the object back to the Server.

Structure Window updates

When you are developing with 4D Client, 4D Server periodically refreshes your 4D Client's Structure Window in the background while you are working. This "flashing" of the structure window can be a bit distracting. To overcome this problem, you should shrink your structure window to a very small size, unless you need to do something in that window.





Compiling

If you are at a remote location and the structure is on a Server in your office, you can install Timbuktu Remote on the Server and on your client machine. Using Timbuktu, you can control the server machine as if you were in the same room; you can launch 4D Compiler on the server machine and compile the structure at normal speeds. Compiling in this manner is just as fast as it would be if you were working directly on the server machine. After the compile, you can copy the compiler's files (Error log, Symbol table, and Type file) to each 4D Client machine.

Advantages of 4D Server development

The advantages of 4D Server development, in most cases, far outweigh the disadvantages. Here is a list of the most obvious advantages:

Simplicity

You never have to worry about which version of the structure, on which machine, is the latest version. You set up a folder on the Server called Current Development Version and keep only the latest copy of the development structure and data in that folder. You have access to this copy when you are in the office, at home or traveling on the road.

Hard drive space

When you are developing an application that will ultimately have a very large data file, you will get to the point where you need to have a large data file available when you are doing the development. For example, when you write search routines, you need to test the code against real data, in order to ensure that the search will return the desired results. Also, you will need to measure the performance of your application against a real data file. If you try to maintain large data files at home and on your laptop, you would be using a great deal of hard drive space for redundant files.

Realistic performance measurement

Perhaps you have experienced this situation: you have worked hard to complete and deliver an application that you are proud of, but the customer immediately complains about performance. You can't understand this, because it was "very snappy" on your machine. But if you are like many developers, you have the latest toys: a Power Mac or Pentium 120mhz with 32mb of RAM, an accelerated video board, and a very fast hard drive. Your customer has a Power Mac 80mhz or Pentium 75mhz server with Quadras and 486s as workstations.

Because you were developing on such a fast machine, you had no idea of what the application's performance would be in the real world. However, if you develop with 4D Client, you will get a much more realistic picture of performance—you will know right away if the application has a performance problem.





4th Dimension

Technical Note 96-38

A good general rule about performance is: strive to make the performance acceptable over a remote connection with a Quadra or a 486. If it performs "OK" under those conditions, it will perform "superbly" in an office over local Ethernet.

Multi-platform testing

If you set up two or more 4D Clients, a combination of Macs and Windows machines, you can immediately see the results of your work on forms and reports. You can design a form on the Mac, go into Runtime mode and see the results. Then on the Windows machine, you can go into Runtime mode, and see the results. If you were developing in single user mode, you would have to Transport the structure over to the other platform, look at your work, make modifications, and then Transport the structure back to the original platform. Therefore you will find that in most cases, for development of a multi-platform application, 4D Server development is much faster than single-user development.

Multi-programmer development

The ability for several programmers to simultaneously develop on the same structure is probably the biggest single advantage of 4D Server development. If you have ever tried to manually merge the code of several developers working on an application, you know that this is a tedious and error-prone process. 4D Server, on the other hand, provides a smooth and seamless system. Whether the programmers are all in the same room or whether they are in different cities; they can all be working from one code base.

Caution: there are three potential disadvantages of multi-platform, multi-programmer development:

1. Access to common objects
4D performs "object locking" on Structure objects, just as it performs record locking on records. Therefore, members of the development team should make it a habit to close layouts and procedures as soon as they are done with them, so that these objects are available to other developers.
2. Respecting another programmer's code
We had some cases in which a developer would change some code that had been developed by someone else, not realizing that this change "broke" a routine that the other developer had written and debugged. This is a potentially explosive situation. In Part 2 of this series, we will show you how to divide a large project into Modules and then assign a different password to each module. This restricted access will minimize the danger that one person or team will override the work of another person or team.





4th Dimension

Technical Note 96-38

3. Window Title Bars moving up and off the screen
This can be a problem when an object is moved to a certain position on one platform (e.g., Mac), resulting in it being too high up on the screen, so the title bar is hidden, on the other platform (e.g., Windows). All developers will have to know the keyboard combinations for moving windows in 4D when the title bar is hidden. These key combinations are:

Macintosh: Command+Control+Click+Drag

Windows: Control+Right Mouse+Click+Drag

Multi-platform documentation

When you are writing documentation and you need screen shots from both Mac and Windows, it is advantageous for you to do 4D Server/4D Client development. If you use 4D Write as your word processing vehicle, you can paste screen shots from Mac and Windows into the same document. For example, you can paste in a screen shot on the Mac, save the record, move the record pointer to the next record; then, go to your Windows machine, open the original record containing that same 4D Write area, paste in the Windows screen shot, and save the record. Now you can view both screen shots on both platforms. This is the fastest way to do multi-platform documentation.

Summary

The traditional way of doing 4D development is to run in single-user mode on a local machine. However, in this age of large multi-platform, multi-developer projects that require development and testing on both Macintosh and Windows, it is wise to consider the option of storing your development structure and data in a central location, and doing most of your development with 4D Client connected to 4D Server.

We will continue our discussion of 4D Server development in Part 2 - Setting Up a Multi-Developer 4D Password System, and in Part 3 - Naming Conventions for Multi-Developer 4D Projects.

