

Chapter 2 - Getting Ready for Success

2.0.0 Luck - A Definition

Yes, it is true that some people are lucky. But did it ever occur to you that luck might be attracted by certain behaviors? Here are a couple of sayings for you to ponder:

"It never occurs to fools that merit and good fortune are closely united."

-- Johann Wolfgang von Goethe

"Luck is that intersection where Preparation meets Opportunity."

-- Anonymous

It has been my personal observation that successful people are lucky. However, I have also observed, from watching them in action, that they work hard at *preparing* themselves for those lucky situations, and they don't hesitate to *"answer"* (take action) when lady luck calls. In other words, they earn their luck. What I am saying is this: if you want to be consistently lucky, you have to earn it. You have to prepare for luck, and you have to take action when lucky opportunities occur.

I want to share with you my theory of luck:

"Everybody has an equal share of potential for good luck. However, those who benefit the most from their potential good luck are those who prepare for good luck, look for good luck, and take advantage of good luck whenever it appears. In that sense, you could say that a person makes his or her own luck."

-- Walt Nelson

As 4D Developer/Consultants, I believe that there are certain simple things that we can do to prepare for good luck. If we do those simple things, luck will come our way - and we will be ready to take advantage of it. If it is true that, as the saying goes, 'Luck is Preparation meeting Opportunity,' then those simple things are the "preparation" part of being lucky. I will list the ten things that I have discovered that can help us to prepare for luck, and then I will discuss each item in detail.

4D for Fun & Profit - Getting Ready for Success

- Practice with the Real Thing
- Become Your Own Customer
- Know Your 4D
- Know the Operating Systems
- Read the Online Discussion Forums
- Read the Manuals
- Join the Partner Program
- Study Database Design Theory
- Invest in Hardware and Software Tools
- Maintain a Positive, "Can-Do" Attitude

2.1 Practice with the Real Thing

Because necessity brought me onto this programming game in the first place (my company needed a database to track our customers and their insurance policies), I stumbled right into this principle of *practicing with the real thing* without realizing that it was a key ingredient for attracting luck in 4D consulting. But I have since realized that having to write and maintain a mission-critical application for my own company was a great training experience.

I installed, upgraded, and supported this application for six years, gradually migrating it from *QuickFile* to *AppleWorks* to *Helix* to *Multi-user Double Helix* and, finally, to *4th Dimension*. All that time, we were using the application for real work: first as a simple customer list, then as a customer list and policy service database, and finally as a customer list, policy service database, mail-merge application, and billing system. Over that period of six years, we experienced every problem you can possibly imagine. You name it, we experienced it. As a result, when I went into 4D Consulting as a profession, I was very much ready to support applications for clients; I knew what to expect.

Let me recount for you some of the experiences that we had in my company, and some of the hard, cold, lessons that I learned from those experiences.

2.1.1 A Lesson about Upgrades

Hot dog! It was 1986, and *Multi-User Double Helix* had just been released. My three-person office had been working with a single-user database. We had been inputting data into three copies of the database and then using "sneaknet" to export/import data. Also, the current version of *Helix* had some irritating bugs that had been fixed in the new version. But perhaps the most exciting thing about the new version was that it would be my first multi-user application. Hot dog!

I sent in my money for the upgrade and waited impatiently for the package to arrive. Every day, from mid-morning until noon, I was lurking at the door waiting for the mailman. This was definitely a case of a "watched pot that seemed to take forever to boil." After 10 days, I called

4D for Fun & Profit - Getting Ready for Success

the company to find out where was my upgrade? Had they already shipped it? When did they ship it? How did they ship it? I was told that they had so many orders for the upgrade that they had been backlogged, but my order had gone out five days ago. I knew that mail to Guam usually took 7-10 days, so I knew that it would be no more than one more week.

Finally, on a Thursday morning, the upgrade arrived. I hadn't been so excited since the time when I got a Schwinn bicycle for Christmas as a kid! I canceled all my appointments for the remainder of the week, took the documentation to a coffee shop, and sat down for five hours of some serious reading.

That weekend, I literally lived in the office: writing multi-user code, converting data, and installing workstation software. Multi-user database computing fascinated me. *Double Helix* was especially fascinating, because it automatically updated the display on all workstations whenever there was a change. I don't know how many times I made a change to data at one workstation, and then immediately ran over to another workstation to watch the screen update itself right before my eyes!

On Saturday afternoon, after the employees had gone home for the weekend, I moved all the workstations to within a few feet of each other so I could watch the automatic screen updates without getting up from my chair.

2.1.2 The Moment of Truth

Monday morning. I had been up until 3:00 am making some last-minute changes, then I was back in the office at 7:15 to get ready for the employees to arrive at 8:00 am. Everyone was excited about going multi-user because it meant we would not have to worry about two people making changes to the same record, and because we wouldn't have to export-and-import data anymore.

It was a disaster. We had at least 12 crashes in the first hour; I stopped counting after that. By the end of the day, we had gone back to the old, single-user version. The data in the multi-user version was so corrupted that we couldn't even export it. We had to re-input the entire day's transactions -- and hope that we hadn't forgotten any. Even now, almost 25 years later, the memory is still painful.

Finally, three upgrades and nine months later, they released a version of multi-user *Double Helix* that was stable enough to use in a live situation. The *Double Helix* fiasco was not my first unpleasant upgrade experience and it was not my last; but it was certainly my most memorable. As a result of that experience, I formulated my *Tried & True Rule*.

2.1.3 Walt's 'Tried-and-True' Rule

Here is a rule that I live by in my 4D consulting life:

"Never, never, ever be among the first to go live at a customer site with a new version. Wait until the version has been successfully installed by many other developers, and until you have done some thorough testing yourself. Wait until the new version is no longer new - wait until it is Tried-and-true."

That's what I call 'Walt's Tried-and-True Rule.' Let me explain further: my definition of 'Tried and true' is a version that has been in use for several months, and has been found to be reliable by many users in many different situations. And I am not referring just to software; I also include hardware in the rule. Never, never, ever place a new, untested computer model in a live environment - especially as a Server. Test it first, read blogs and the experiences of real-life users. No matter how fast and wonderful the new model may be, test it yourself for a period of days or weeks, and sign on to online forums to find out what experiences others have had with that model.

Back to the Helix story: at the time that it happened, the Multi-user Helix fiasco seemed like my worst nightmare. Looking back, however, I realize that it was the best thing that could have possibly happened, for two reasons:

1. I learned a valuable lesson that has kept me out of "hot water" hundreds of times in my 4D consulting career.
2. It happened in *my* office with *my* employees. I hate to think what would have happened if it had a client's office and a client's employees. That could have been the end of the relationship with that client. At the very least, it would have put a tremendous strain on the relationship.

And with that, we get back to the point that I am making with this true story: Before you go into 4D consulting, you would be wise to **Practice with the Real Thing**: gain some experience working on real-life applications for your employer or for yourself.

2.2 Become Your Own Customer

You may be wondering: "What if I am already a 4D Consultant, or what if I am not fortunate enough to be able to do 4D work for my employer, and I don't need any software for myself?" Well, my friend, that's impossible. You will always have database needs of your own: your stereo or video collection, your correspondence -- and, after you go into 4D consulting, your customer list and your invoicing or time-billing system.

I can think of only one good reason for a professional 4D developer not to write his or her own database for his contact management, correspondence, time billing, or invoicing: lack of time. On the other hand, I can think of several good reasons why that same 4D developer *must* write his own contact management, correspondence, time billing, and invoicing system. As a professional 4D consultant/developer, you need a 4D application with which to:

- Test new releases of 4D;
- Test new concepts about 4D programming;
- Test new or upgraded 3rd party add-ons

If you don't have time to start your own database from scratch, then you should get a basic system or "shell" from 4D or from another developer, and modify that structure to suit your own needs. In addition to handling invoicing and correspondence in this database, you

4D for Fun & Profit - Getting Ready for Success

should also use it as a training ground to improve your 4D programming skills. So you should constantly be adding new "bells and whistles" to your database.

Just in case you're still not convinced, let me give you one more good reason for setting up your own database and becoming your own customer: it's an old rule of sales that:

"It is much easier to sell a product that you believe in."

If you have your own 4D database, and that database is doing good things for you, you will have first-hand knowledge of the practical value of 4th Dimension. As a result, you will sell your services with more confidence and self-assurance because you believe in the product. Trust me; this is just as true for 4th Dimension as it is for insurance, or detergent, or health foods.

I practice what I preach: I have four databases that I use for myself and my businesses, which are maintained in 4D:

- A video collection of more than 1,200 VHS, DVD, and Blu-ray's
- A Real Estate School database which handles generating questions, sending class materials out to the web, web grading of student exams, tracking attendance, and tracking revenues.
- A Document-generation database which I call *Docupower*. It uses 4D Write and Hierarchical Lists to manage documents with re-usable text, graphics, and book preferences. I am writing this book in Docupower, and some of the examples that I will give you later in this course were written using Docupower.
- A time-billing application that tracks clients, projects, and hours

The most recent version of 4D, at the time of this writing, is 4D Version 12.1. It appears to have reached a level of stability, so I am in the process of converting all four of my databases to Version 12.1.

Here is another rule that is a corollary of the Tried-and-True Rule:

Key Point: Walt's Rule of Self-Testing

"Set up your own databases to track things that you need to track. Try out all your new ideas all new versions of the Operating System(s) you support, and all new versions of 4D on your own databases. Never, never, ever install at a client site a new version of 4D, 4D Server, a 3rd Party Add-on, System software, or any new model of hardware in a client site unless you have extensively tested that hardware or software yourself, and unless the people that you respect in the 4D online forums have given that new software or hardware a passing grade."

You may think this rule is pretty extreme, that it is over-kill. But believe me, if you don't follow this rule, you will be burned many times by situations like my *Double Helix* fiasco. So, even though a client may be pushing you to install the latest toys, you need to learn to

resist - to tell the client that you want to wait until that new toy is not so new; until it is Tried-and-True.

2.3 Know Your 4D

The previous discussion of setting up your own database leads quite naturally into the next step in preparing yourself to take advantage of the good luck that will inevitably come your way as a 4D consultant: Know Your 4D. This seems obvious, but it is the obvious things that we sometimes overlook. So please indulge me while I remind you to Know Your 4D. I have seen many, many cases - as the Director of Tech Support at 4D Inc. and in online forum discussions - where developers have reported spending hours and hours trying to do something that they could have done easily, if only they had a better knowledge of the commands, functions, and design techniques of 4th Dimension.

If you are really serious about 4D consulting, your goal should be to learn every one of the commands, functions, and tools available in 4D. Since 4D is constantly evolving, this means you should never stop studying, never stop learning about the 4D universe.

I guess I am lucky on this, because I'm the type of guy who reads the instructions before using a product; reading the 4D manuals comes naturally to me. Whenever a new version comes out, I read the manuals from cover to cover several times. When I joined the Silver Surfer beta test program, I read the entire two volumes of documentation during the first week that I was experimenting with the product, and then I read all the manuals again when 4D was officially released. And each time there has been a major upgrade, I have read every manual from cover to cover. In fact, when I was working at 4D Inc., I actually wrote the 4D Quick Start Manual.

Whenever a new version comes out, I put at least one of the manuals in the bathroom - or take my computer into the bathroom - and use the manual as "library reading." Don't laugh - it works for me!

2.4 Know the Operating Systems

Another essential for attracting good luck and success in 4D development is that you must know the Operating Systems that your clients are using. In many cases, your clients will need you to be their "operating system guru," especially when they are just getting started. For you, as a 4D developer/consultant, knowing the operating systems that your 4D applications run on is nearly as important as knowing 4D itself. If your clients are on Macintosh, you need to know the Mac OS; if your clients are on Windows, you need to know the Windows OS; if your clients are on both, you need to know both; and if your clients are running different versions of the OS's, you need to know all the versions.

My own experience as a 4D consultant has been that a significant percentage of my trouble calls - in some cases as much as 50% - have *not* been about my 4D application; these calls have been about problems with the other pieces that make my application useful to the

client: printing problems, networking problems, hardware problems, and operating system stability problems.

Key Point:

As a true professional, you should become one of the few people on earth who has actually read the Readme documents, the operating system manuals, the printer manuals, and the network card manuals.

2.5 Read the Online Discussion Forums

If I had to choose one thing that I definitely would *not* give up, it would be the 4D discussion forums online. One of the best things that you can do to get ready to become - or to enhance yourself as - a professional 4D Developer is to subscribe to 4D discussion groups on, and to faithfully read the message traffic every day.

After you subscribe, you should become very active - at least to the point of reading a large percentage of the traffic. (Every day, I read *every message* on the 4D iNUG, and I have done so for many years.) I have been going online daily for most of my 4D career: First there was Compuserve, and now the Internet. In the early days, there was a one-year period when Compuserve added a \$50.00-an-hour surcharge for Guam, but I continued to sign on anyway, because I learned so much about 4D from reading the traffic that it was worth it. The only thing that I cut back on was my downloads of large files (e.g. example databases). Nowadays, we have monthly unlimited access to the internet at a flat rate. Such a deal.

What are the benefits of reading 4D discussion group message traffic? The number one benefit is that you get nearly-immediate answers to questions and problems. Many times, I have seen someone post a question online, and then get four-to-seven good answers within forth-eight hours. Not only that, but several dozen other people probably read about the problem and the solution. Perhaps they archived the information for future reference. Another benefit is that every time there is a new version of 4D, of a development tool, a plug-in, or a 3rd party product is released, people go online and post their observations, problems, and questions. If you monitor the discussion groups for the first few weeks after the release of a new version, you will have a very good idea of how solid the release is, and you will know exactly what bugs have been encountered by others. That alone, for a 4D consultant, is worth at least 10 times the cost of online access!

2.6 Read the Manuals

I already mentioned that I always read the manuals from cover to cover, but I want to spend some more time on this point because it is so important, and because I have learned, over the years, that some developers do NOT read the manuals.

Several few years ago, when discussing the idea of reading the manuals with a fellow 4D developer, he point out that one reason that people like the Graphical User Interface of the

Mac is that they can use the software without having to read the manual. I agreed that is true, but programmers are not normal people. We programmers are the ones who should make the programs easy for normal people to use. In order to do that, we have to study, study, study. He disagreed with me at the time; however, after struggling with 4D for over a year and watching me pull way ahead of him in 4D knowledge and expertise, I could tell that he started reading the manuals, because he started to exhibit a knowledge of the commands and functions of 4D -- a far more in-depth knowledge than he had before.

There are so many times when you need to do something in 4D, and just knowing that a command or function exists can make the difference between getting the task done in a few minutes, and struggling with it for hours.

Let me clarify one thing: with all the hundreds of commands and functions in the 4D universe, the development tools, the plug-ins, the components, and the 3rd party add-ons, I do not expect you to memorize all the syntax. But what you should try to do is know what commands are at your disposal. Then, when you need to use a command, you can look up the syntax.

2.7 Join the Partner Program

Another way that you can attract good luck in your 4D development efforts, is to join the 4D Partner Program. Some people feel that the 4D Partner Program is not worth the cost; however, 4D now has three levels of Partnership (Bronze, Silver, and Gold), starting at just \$399 per year. Depending on your organization's needs, you can upgrade your Partner membership to get extra user ID's, extra test servers, extra 4D Professional licenses, and so on. In short, there is a membership to fit every budget. But it is only fair to tell you that I am biased - and here are the reasons why:

- I was a member of the program for several years before I became an ACI employee. (Note: ACI Inc. was the name of the company before it was changed to 4D Inc.)
- My first position at ACI was "Manager of Partner Services" which means that I managed the team that supported the ACI Partners Program.
- My next position at ACI was "Director of Technical Support" which means that I managed the team that provided technical support to to customers. Most of our calls were from ACI Partners.
- My final position at ACI was "Director of Partner Development" which meant that I provided high-level training to existing partners, and I actively recruited new members to the program. As of this writing, that was about 10 years ago; and for most of that ten years, I have continuously been a member of the program.

I don't want you to join the program just because I said you should. You have to make your own choice as to whether the program is worth it to you. But if I may, I'd like to offer this

4D for Fun & Profit - Getting Ready for Success

suggestion: look at the benefits you are gaining from the program and ask yourself whether it is worth at least triple what you are paying. My philosophy is that if I'm getting my money's worth and then some, it's almost irrelevant what others think I "should" do. One of the secrets to success in life, in my opinion, is learning to think for yourself and learning to act according to your own hunches, regardless of what other people say, think, or do. So I do not want to influence you too much in this decision. But I will tell you the benefits of the Partner program that are most important to me (Through my client, I am at the Silver level - \$1,299 per year):

- Test Licenses: 4D Developer Professional, 4D Server, 4D Remote (two licenses), 4D Write, 4D View, 4D Web Expansion, 4D Web Services Expansion.
- Beta Versions: I like to test out new versions long before they are released, so I can plan way ahead. Having access to Beta versions is important to me; it gives me a four-to-six month head start on knowing and understanding the features of every new version.
- Hotfixes: once in awhile, there is a show-stopper bug that I need a fix to, and I need it before the next formal release. 4D will often fix these bugs in Hotfixes, and release them to registered Silver and Gold Partners.
- The Latest Technical Notes (TN's): when they are released, tech notes are only available to Registered Partners for the first 12 months; others cannot get access to the TN's until after 12 months.
- 4D Partner Central: this is a web site containing additional technical resources and code examples.
- Technical Support: as a Silver Partner, I get twenty-four (24) "incidents" per year. To be honest, I have never used all 24; I might only call 3 or 4 times a year. But when I call for help, I *really* need help.
- Bugbase: access to all the reported bugs for all currently-supported versions.
- Silver and Gold partners get discounts on 4D Training, and on the annual 4D Summit.

2.8 Study Database Design Theory

To attract good luck in your 4D development efforts, you need to study classic Database Design Theory.

Will Rogers said: "Everybody is ignorant - only on different subjects."

If you (like me) did not get into the 4D consulting business through the route of a Computer Science degree, and if you (like me) did not have a background in working on large computer systems projects before becoming a 4D developer, then you started your 4D career

4D for Fun & Profit - Getting Ready for Success

ignorant about database design theory. Fortunately, ignorance is a curable disease. And the most important step in any cure is the diagnosis: the realization - no, the admission - that one is ignorant.

When I began 4D programming, I was guilty of a kind of reverse-snobbery, actually believing that the "old way" was useless; database design books were useless; the "new way" was to just dive in and prototype one's way to heaven.

Fortunately, I had a guardian angel around me, in the form of John Foster, who patiently guided me to a knowledge of "the way." John helped me to see that just because databases were moving away from mainframes/minicomputers and into micro-computers, that didn't mean that everything they did in the mainframe and minicomputer world was obsolete. Some principles endure. John passed that wisdom to me, and now I pass it along to you: if you learn the secrets of the ancient mainframe database high priests, you can find database nirvana. (Just a bit of humor here, but humor with a serious message.)

Get into the books and study database design theory and "life cycle management." You will find that much of what they teach is still relevant today. One word of caution, however: when you start studying database design and life cycle management theory, you can take it too far. Don't go overboard and spend all your time memorizing the the definitions of impressive-sounding terms; the main thing that you need to learn, in my opinion, is how to take a real life information-storage problem and turn it into a workable database design.

One of the most important techniquet that I have found useful for creating a workable database design is a technique called Normalization. I call this process "reverse-engineering" because that is, in effect, what we do in the Normalization process.

The Normalization Process

- Step 1: Take the outputs (e.g. the invoices and reports) that the client wants, and work backward from those documents to create a database design: Tables, Fields, and relations among the tables. This is the actual Normalization of the database. However, we don't stop here. We continue as follows:
- Step 2: Using that database design, prototype the input screens that will handle the operator input;
- Step 3: From those input screens write pseudo-code that will do the batch updating the produce the reports;
- Step 4: Knowing the target platform (Macintosh, Windows, or both), implement the input screens and turn the pseudo-code into real code.

That, in a nutshell, is the process of Normalization: designing and constructing a database through reverse-engineering. Each step in the process has a large stack of books written about it. Go to your local book store, pick out a couple of books that cover the entire process, and read them. You will be well on your way to knowing

how to effective database design. And if you get stuck and need advice, the most obvious place to turn is the online 4d community.

2.9 Invest in Hardware and Software Tools

Another way to attract good luck as a 4D developer is to acquire the appropriate hardware and software tools. As any good mechanic will tell you, there is no substitute for having the right set of tools. In 4D Consulting, your "tools" are your hardware and software. In the next several pages, I will make some suggestions as to the minimum configuration that you should have, and why.

2.9.1 Hardware

Here is a list of your minimum requirements for hardware. This list is not all-inclusive; in your particular practice, you may find that you need tools that are not on my list.

- A laptop or desktop computer with a large hard disk, as much RAM as it will hold, an Ethernet card, and - if it is a laptop - a wireless modem.
- If you support both Macintosh and Windows, then you should have a computer for each platform, or run both platforms on your Macintosh.

Note: from 1984 until 2004, I was a die-hard Macintosh fanatic. However, in 2003, I noticed that none of my clients owned a single Mac! Gradually, they had all made the transition to Windows. I haven't bought a Mac since 2002, and my last Mac stopped operating in 2008. So I cannot give you any advice on how to set up "Parallels" or any of the other tools that allow you to run Mac and Windows on one machine.

- A writeable DVD drive to use for historical backups of your development work. At least twice a month, I make a copy of all the current structures I am working on, and I put that DVD away in a safe place, "just in case..."
- High-capacity Flash Drive: you should have at least one USB drive ("Flash Drive") that you carry around with you. I have my USB drive attached to my key chain. I use this drive to transfer files to and from client systems, and sometimes to make temporary backups.
- Printers. We have several laser printers in my office, and I have a small laser printer at home. If you sometimes need to print documents on the road, you should also consider getting a portable printer.

2.9.2 Software

Here is a list of your minimum requirements for software. This list is not all-inclusive; in your particular practice, you may find that you need tools that are not on my list. And keep

4D for Fun & Profit - Getting Ready for Success

in mind that if you support clients on both Macintosh and Windows, you should have a full set of software tools for each platform.

■ **The 4th Dimension Family**

- 4D Developer Professional. This includes 4D Single User, 4D Web Server for development and testing, 4D Web Services Server for development and testing, 4D ODBC Pro, 4D Write, 4D View, a SQL Server for development and testing, and the 4D Unlimited Desktop application builder.
- 4D Server with two-user license. Ideally, you should have the same plug-ins attached to your 4D Server as you have attached to the 4D Developer Professional, but this can be expensive - unless you are a Silver member of the Partners Program, in which case you get a 4D Server license - with all the goodies attached - as a part of your membership.
- Free 4D Components such as 4D Pop

■ **Third-Party Add-ons and Utilities**

Depending on your particular 4D "practice," you might find a need for one or more of these utilities and plug-ins:

- *DataCheck* for doing integrity checks of a client's data file and indexes (a must-have, in my opinion. More on Datacheck later in this book.)
- *Foundation Shell* as a starting platform for new 4D projects

■ **Productivity Software**

The following software helps me be more productive. Note that I am now Windows-only, so this list only includes software products that are available on Windows. If you are on Mac, you can either run the Windows version in a virtual Windows OS, or you can find a Macintosh equivalent.

Software	Description
Web Browsers	MS Internet Explorer, Google Chrome, Firefox Mozilla
Microsoft Word	Word Processing
Quicken	Personal Finance
Microsoft Excel	Spreadsheet
Microsoft Visio	Flowcharting (I will say much more about this later)
Wordpad	Simple word processing
Avira Antivir	Virus Protection
Greenshot	Screen capture/editing utility (excellent freeware)
Winzip	Disk file compression

■ **Operating System Software**

As new versions of operating systems are released, you should be among the "first on the block" to try them out for yourself. (The smart way to do this is on a spare machine that is not mission-critical for your development activities, or in a separate partition on your main development machine.)

4D for Fun & Profit - Getting Ready for Success

If there are new features that will make your job easier, you need to know about those; on the other hand, if there are fatal bugs and system crashes associated with the new version of the O/S, you need to know about those too.

It is only a matter of time before our clients have copies of the new operating system installed on their machines, so you should become familiar with every new O/S as soon as possible. However, note the following caution:

Key Point

For the first few months after a new operating system is released, you should NOT voluntarily install it at a client site - because of the Tried and True Rule. In some cases you will have no choice; for example, when the new version of the O/S comes pre-installed on a new computer. However, when you DO have a choice, stick with the tried-and-true rule.

2.10 Maintain a Positive, "Can-Do" Attitude

To attract good luck as a 4D developer, it is imperative that you to maintain a positive, can-do attitude.

Yes, I can hear you thinking: "Oh no, not another lecture on Positive Thinking." Yes. I am giving you a short lecture on staying positive in this chapter; the 'extended version' will come in the *Fishing Poles 401* chapter. But in the meantime, I want to share the following with you:

Key Point

Having a belief that something can be done; that you can do it; and that you can do it now; is an essential pre-requisite to success in all things. Some people have trouble believing that they can do something because they don't know how - step by step - they will accomplish the thing. Don't let your limited knowledge of the "how" stop you from believing in the "what." Don't worry about how it will be done, just get started. As you take each step along the path, the next step - the next "how" - will be revealed to you.

I am not the first one to say this; the same message has been repeated thousands of times over the millenia. There is an old Chinese proverb:

"A journey of a thousand miles begins with the first step."

2.11 Not All at Once

Let's quickly review the luck-attracting actions that I have recommended to you in this chapter:

4D for Fun & Profit - Getting Ready for Success

- Practice with the Real Thing
- Become Your Own Customer
- Know Your 4D
- Know the Operating Systems
- Read the Online Discussion Forums
- Read the Manuals
- Join the Partner Program
- Study Database Design Theory
- Invest in Hardware in Software Tools
- Maintain a Positive, "Can-Do" Attitude

Remember that the above is your long-term "wish list." You may now know how you will accomplish all of these things right now, but that's OK. Just get started; take the first step: pick one thing on the list that you feel is the most vital right now, and get started on it. Work on that one thing until you are satisfied that you have made significant progress, then begin on the next. As you go through the list and take the first step toward accomplishment of a goal, each new step will reveal itself to you. In a year, you will have made significant progress on the majority of the items on the list.

That's pretty good; it took me over 5 years.