McLEAN MOBILE COMPUTING CRYSTALS

Email: mike@mmc2.com.au Mobile: 0403 409 961

October 2020

# MOBILE COMPUTING REVOLUTION

First there was the **computer revolution**. This introduced the term Electronic Data Processing. The term 'revolution' referred to the quantum change in the way that information was handled. Hand written information, and pieces of paper  might get the job done, but there was now a **better** way. This fact is sometimes highlighted when we are in the bank or some office, and are told that the computers are down. Our expectations have moved on to such an extent, that even being told that the computer system is slow, brings groans from the waiting customers.

Within the computer revolution, which is still going on, we can identify a number of sub revolutions. These include the massive changes that occurred when computers no longer required a computer room with air conditioning, and masses of fluorescent lights, but rather sat on your desk. The same basic elements of the computer - a processor, memory and a software program, could now be arranged in ways that meant new applications could be efficiently computerised. There was no reason why these applications could not have been computerised before the advent of desktop computers, at least not from the computer side of things. But for something to be **feasible**, we must take into account more than whether the computer program can be written. We must take into account the people using the computer, and the context in which it is being used. The reduction in both the physical size, and the stringent requirements, **changed the context** that the new type of computers were used in, <u>and this is the reason</u> why they were able to be **efficiently used** in areas that had not been previously applicable. Perhaps the idea that Desktop computers, often called PCs (Personal Computers), was a sub revolution of the greater Computer Revolution, is not accurate in some ways. In many ways the use of PCs has far outstripped what we think of as the 'Original' computer revolution.

The advent of Laptops and/or Notebooks further extended the relentless forward march of computers. And somewhere along the way, commentators realised that it is not so much the computer itself, no matter how powerful and impressive it may be, but how we use it, that really counts. This realisation brought forth the concept, not so much of the 'computer revolution', but gave birth to the new term, the '**Information Revolution**'. **The computer is the tool,** but the use, or **the application,** of the computer **is where it is all at.**

A major part of the 'computer package' is the Operating System. First there is the physical computer itself, or the hardware, then the operating system, e.g. Windows, and then there is the application, or the particular computer program, maybe a game, maybe a word processor, maybe a business program, maybe a sports scoring program. A good computer operating system should have similarities to a sports referee. Both should do their jobs, and do them well, but in so doing have a degree of invisibility, or in other words stay in the background, only becoming obvious when needed. But for some reason, **operating systems became more and more prominent,** asking, in fact **demanding** to be replaced by newer and 'better', and costly ones. **Operating Systems are also demanding of our time and**

**attention,** requiring someone who understands how to deal with the operating system. In earlier days this was a specially trained computer operator.

Games, not business programs, require the most resources on computers. This may provide an argument for more sophisticated operating systems for gamers, but for many users there is no justification for having to buy a new operating system every couple of years. For the many users that just want to check their emails, browse the internet, do a bit of social media, and maybe some word processing and spreadsheets, there is no need for more powerful operating systems. This type of usage should be settled and stable by now, and not require  updates and constant later versions. These are more in the interests of the software vendors, rather than the computer users.   I recall a prominent operating system that had printed on the box, that one reason to buy it, was because **it had less bugs in it than the previous version**. What other products would get away with this. It is enough to turn some people off computers. I can speak from personal experience.

There were also growing grumbles of dissatisfaction among  many computer users. Why was it that your computer would slow down over time and develop issues to the point where you would purchase a new computer with all the latest software, hoping to solve these issues once and for all, only to find a similar situation occurring again within a relatively short space of time.

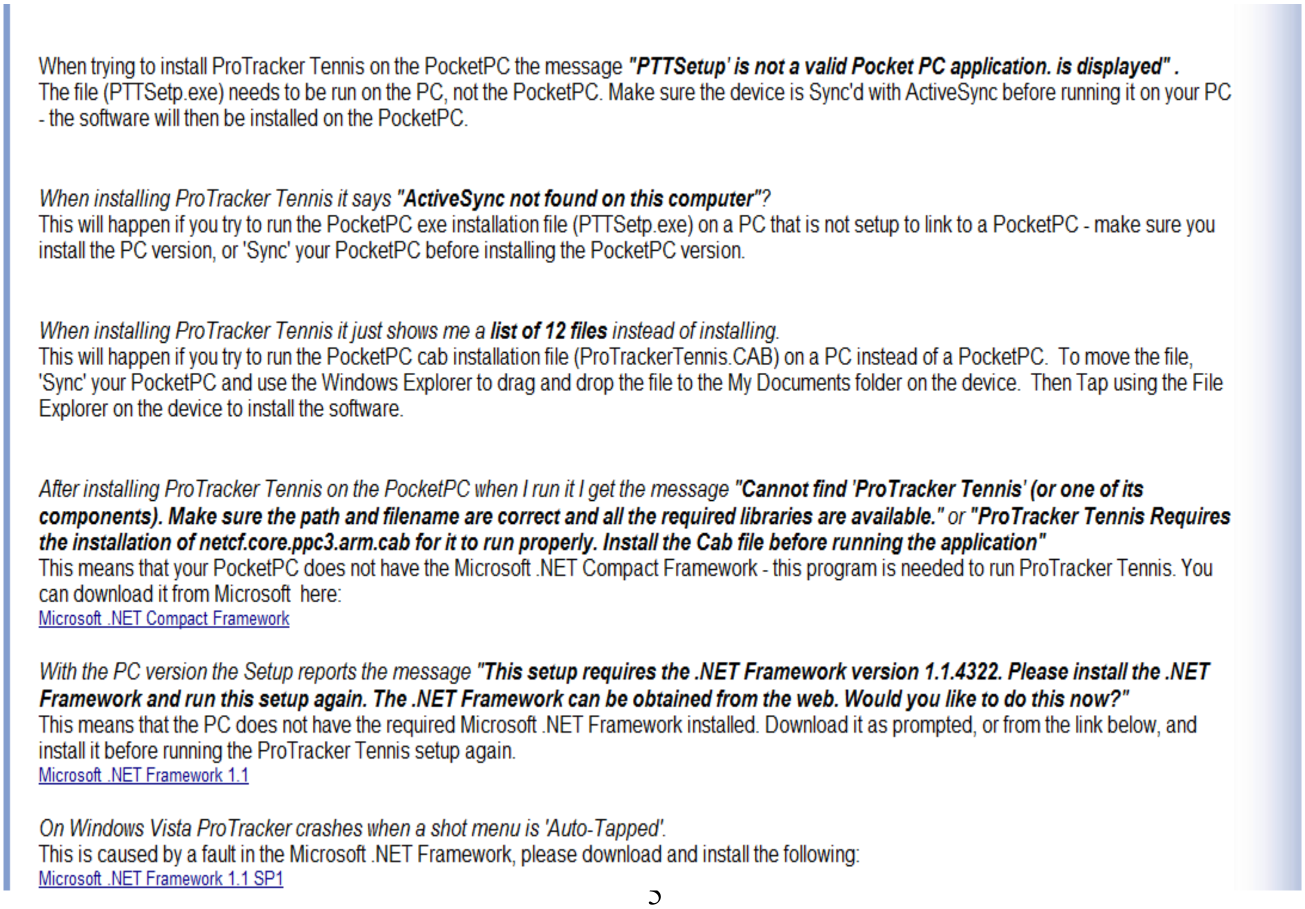
An electronic calculator is a specialized computer, but a computer nevertheless, with hardware and software components. Calculators don't need updates and don't have issues develop over time. Surely the computer industry was mature enough for general purpose computers to be as robust as calculators. But no. **It was time for the context to change again**. This time the revolution, and we are at the beginning of a massive new phase of the computer revolution, is called  **MOBILE COMPUTING**.  Computers became smaller yet again, initially with Palmtops and PDAs and then Smart Mobile Phones. But it is about much more than size. An aspect of this change, and it is something that is a fundamental principal **of the design philosophy behind MMC2 software** applications, is known as the **DEVICE APPROACH**. Palmtops and PDAs and Smart Phones have operating systems as do PCs. At first, software systems analysts (designers) sought to rewrite existing software to take account of the physically smaller and more size restricted screens, and then to make this software available on smartphones etc. **But software designers who have a genuine mobile design philosophy are not content with that**. MMC2, design and supply genuine Mobile Software Applications based on the **Device Approach**.

The 'Device Approach' can be illustrated by contrasting (1) using an electronic calculator, with (2) using a calculator app on a PC, to do some calculations. With the calculator - the device, you turn it on and use it. It is relatively simple and intuitive. You are not faced with the many other, sometimes confusing steps, that you have to navigate on the PC to achieve the same calculations. You do not have to worry about downloading or installing the app, or software updates. By choosing the calculator to do the job, you are leaving behind, waiting for the computer to boot up, and the many annoying, time wasting, and sometimes confusing messages and interruptions that can appear, and pop up on your screen, presumably requiring your understanding, and some sort of response.

The Device Approach means that you have a particular object and it does a job. With the device, you are in the hands of the software engineer who designed your device specifically for this task. This is in contrast to being in the hands of Windows, along with all of its general things to attend to, along with interruptions and popups etc. etc. Your involvement is kept to what is relevant and very easy to understand. This is the central feature of the Device Approach, and it is a key component of the design philosophy behind MMC2 mobile computing crystals.

One way of quickly getting to the 'nuts and bolts' of a situation is to have a look at the **Frequently Asked Questions**, if there is such a section. An example of the differences to the end user, between the two approaches described above, can be seen by looking at the FAQs for a couple of programs. Following are samples from the FAQs for two tennis statistics programs that run on Pocket PCs and PCs. These examples were noted some time ago, and may not be currently relevant to these programs, but they illustrate the point of what mobile computing crystals avoid. The programs are ProTracker Tennis and TennisStat. This is not intended to be critical of either of these programs. The point being made, is the contrast in complexity, in this example concerning system setup, between software designed to operate with the conventional approach, and mobile computing crystals designed with the **DEVICE APPROACH**. While this example concerns system setup, the differences are by no means confined to system setup.

An example of the frequently asked questions for ProTracker Tennis. This is not to suggest that it is not a good program, but to show the possible complexity of 'computer approach' applications when compared with 'device approach' applications.



When trying to install ProTracker Tennis on the PocketPC the message ***"PTTSetup' is not a valid Pocket PC application. is displayed" .***
The file (PTTSetp.exe) needs to be run on the PC, not the PocketPC. Make sure the device is Sync'd with ActiveSync before running it on your PC - the software will then be installed on the PocketPC.

*When installing ProTracker Tennis it says "**ActiveSync not found on this computer**"?*
This will happen if you try to run the PocketPC exe installation file (PTTSetp.exe) on a PC that is not setup to link to a PocketPC - make sure you install the PC version, or 'Sync' your PocketPC before installing the PocketPC version.

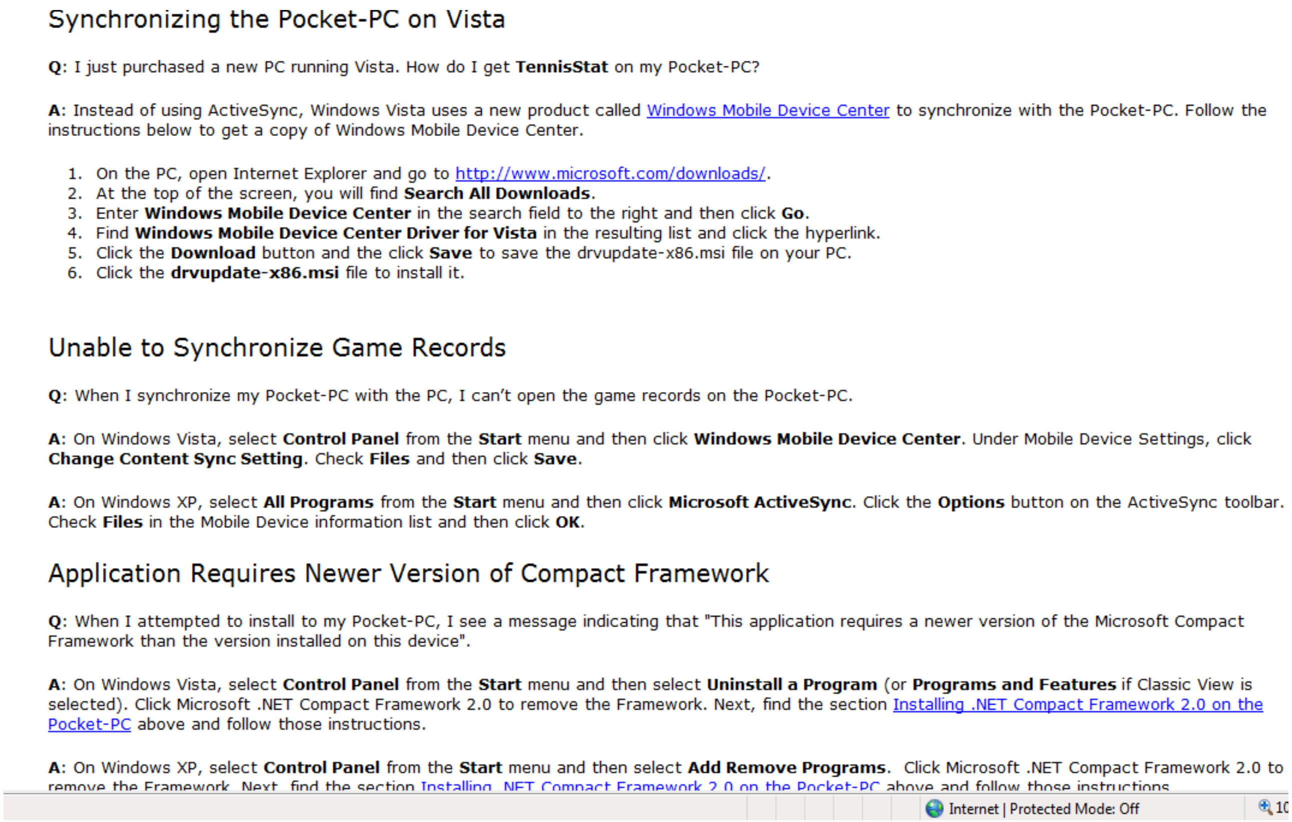
*When installing ProTracker Tennis it just shows me a **list of 12 files** instead of installing.*
This will happen if you try to run the PocketPC cab installation file (ProTrackerTennis.CAB) on a PC instead of a PocketPC. To move the file, 'Sync' your PocketPC and use the Windows Explorer to drag and drop the file to the My Documents folder on the device. Then Tap using the File Explorer on the device to install the software.

*After installing ProTracker Tennis on the PocketPC when I run it I get the message **"Cannot find 'ProTracker Tennis' (or one of its components). Make sure the path and filename are correct and all the required libraries are available."** or **"ProTracker Tennis Requires the installation of netcf.core.ppc3.arm.cab for it to run properly. Install the Cab file before running the application"***
This means that your PocketPC does not have the Microsoft .NET Compact Framework - this program is needed to run ProTracker Tennis. You can download it from Microsoft here:
Microsoft .NET Compact Framework

*With the PC version the Setup reports the message **"This setup requires the .NET Framework version 1.1.4322. Please install the .NET Framework and run this setup again. The .NET Framework can be obtained from the web. Would you like to do this now?"***
This means that the PC does not have the required Microsoft .NET Framework installed. Download it as prompted, or from the link below, and install it before running the ProTracker Tennis setup again.
Microsoft .NET Framework 1.1

*On Windows Vista ProTracker crashes when a shot menu is 'Auto-Tapped'.*
This is caused by a fault in the Microsoft .NET Framework, please download and install the following:
Microsoft .NET Framework 1.1 SP1

Now lets have a look at the frequently asked questions for TennisStat. Again this is not to imply that there is anything wrong with the program but to highlight the possible complexities of a 'computer approach' program when compared with a 'device approach' program.



### Synchronizing the Pocket-PC on Vista

**Q:** I just purchased a new PC running Vista. How do I get **TennisStat** on my Pocket-PC?

**A:** Instead of using ActiveSync, Windows Vista uses a new product called Windows Mobile Device Center to synchronize with the Pocket-PC. Follow the instructions below to get a copy of Windows Mobile Device Center.

1. On the PC, open Internet Explorer and go to http://www.microsoft.com/downloads/.
2. At the top of the screen, you will find **Search All Downloads**.
3. Enter **Windows Mobile Device Center** in the search field to the right and then click **Go**.
4. Find **Windows Mobile Device Center Driver for Vista** in the resulting list and click the hyperlink.
5. Click the **Download** button and the click **Save** to save the drvupdate-x86.msi file on your PC.
6. Click the **drvupdate-x86.msi** file to install it.

### Unable to Synchronize Game Records

**Q:** When I synchronize my Pocket-PC with the PC, I can't open the game records on the Pocket-PC.

**A:** On Windows Vista, select **Control Panel** from the **Start** menu and then click **Windows Mobile Device Center**. Under Mobile Device Settings, click **Change Content Sync Setting**. Check **Files** and then click **Save**.

**A:** On Windows XP, select **All Programs** from the **Start** menu and then click **Microsoft ActiveSync**. Click the **Options** button on the ActiveSync toolbar. Check **Files** in the Mobile Device information list and then click **OK**.

### Application Requires Newer Version of Compact Framework

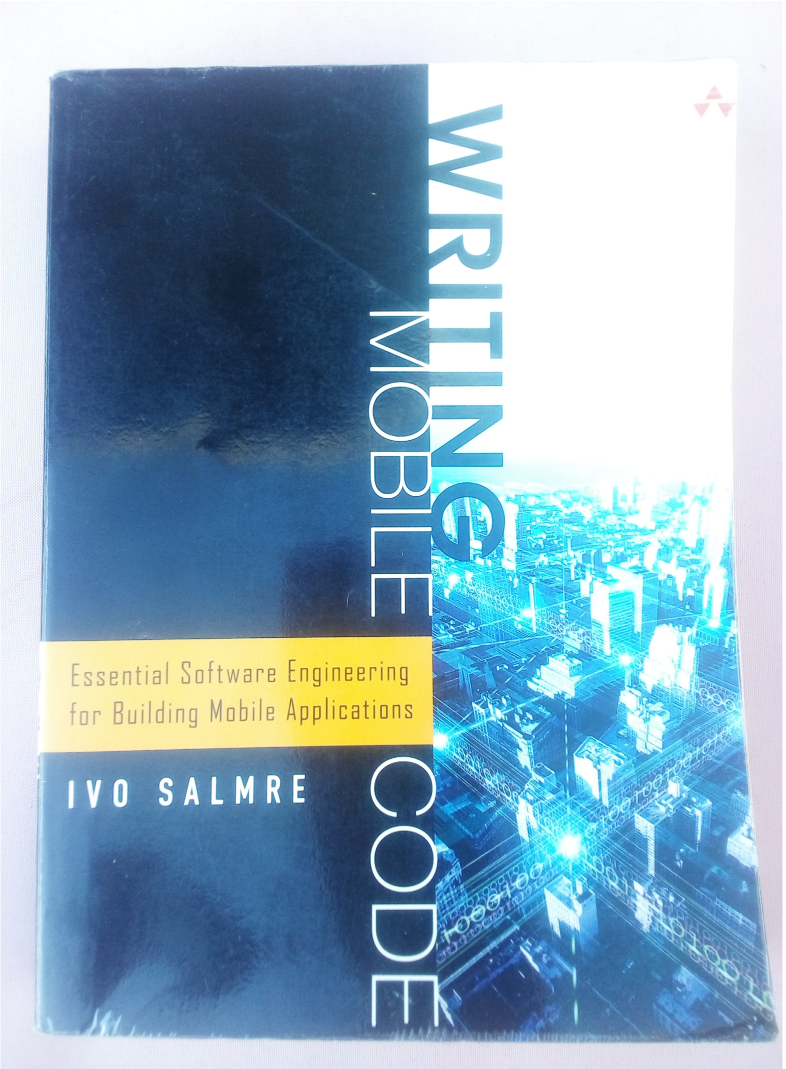
**Q:** When I attempted to install to my Pocket-PC, I see a message indicating that "This application requires a newer version of the Microsoft Compact Framework than the version installed on this device".

**A:** On Windows Vista, select **Control Panel** from the **Start** menu and then select **Uninstall a Program** (or **Programs and Features** if Classic View is selected). Click Microsoft .NET Compact Framework 2.0 to remove the Framework. Next, find the section Installing .NET Compact Framework 2.0 on the Pocket-PC above and follow those instructions.

**A:** On Windows XP, select **Control Panel** from the **Start** menu and then select **Add Remove Programs**. Click Microsoft .NET Compact Framework 2.0 to remove the Framework. Next, find the section Installing .NET Compact Framework 2.0 on the Pocket-PC above and follow those instructions.

**<u>The situations described in the above two excerpts, are simply not applicable to McLean Mobile Computing Crystals</u>.**

I would now like to quote at length from a computer software design book to **reinforce some of the claims that I have made**. The book is called 'Writing Mobile Code', subtitled 'Essential Software Engineering for Building Mobile Applications', written by Ivo Salmre and published by Addison-Wesley 2005. The quotes are from the preface and from the early part of the book and have been 'cut and pasted' to summarise and to suite my purpose. I believe that nothing has been put into a different context or misrepresented.


WRITING
MOBILE CODE
Essential Software Engineering for Building Mobile Applications
IVO SALMRE

Emphasis by use of bold type is mine.

<u>Beginning of quotes</u> “

**There is a mobile device revolution underway.** ...

This book is a software engineering guide focusing on mobile application development because good software engineering is so important for making great mobile applications. It is a topic that has not been addressed yet, and the lack of clear guidelines and techniques is responsible for a great deal of frustration both for developers attempting to move to mobile development as well as for **end users who bear the brunt of software developers' design mistakes**. ...

It is difficult to overestimate the impact truly mobile and ubiquitous computing will have on the way we work, the way we live, the way we communicate, and the way we interact with the world around us.
**Intelligent mobile device software is central to the sea change that is underway.** ...

mobile devices *are* different. **Simply running a desktop operating system and applications on a mobile device will not produce a**

**satisfactory experience for the end user**. As anyone using a modern mobile phone or PDA is familiar with, the device they are holding is indeed a rich computer, but it is in significant ways different from their desktop or laptop computer. The design priorities and user expectations for mobile devices differ from traditional personal computers. ...

Successful mobile applications **bring a highly focused experience to their users, enabling them to zero in on and quickly accomplish desired tasks**. ...

Mobile device applications tend also to be more focused on enabling a few specific features very well as opposed to offering the general-purpose exploratory environment that successful desktop applications do. Because mobile devices are often operated using a single hand ... it is important that users of the device be able to quickly discover and navigate to the information and features they want. **The ability to quickly navigate a small set of key features is an important aspect of a great mobile device experience.** ...

A common mistake developers make when trying to bring a desktop application to a device is trying to bring down the whole application and shoehorn it in to a device. This never works satisfactorily because of the usage **differences** between desktops and mobile devices ...

there is **a huge gulf** between “**being able to write code**” and “**being able to build great applications.**” The latter challenge represents the inherent difficulties of modern software development. It is one thing to be able to sit down, write, and debug code and quite another to knit all the code into a well-performing, reliable, and flexible application.

“ <u>end of quotes</u>.

It would be a mistake to say that mobile computing is just about running the same programs, maybe cut down a bit, on physically smaller hardware. While a cursory glance may make it appear so, nothing could be further from the truth. If the end user is to maximise the advantages of surfing the wave of the mobile device revolution, **it is essential that the software engineers who design and craft the products are not only familiar with, but enthusiastic for the new approach -** the **DEVICE APPROACH**.

All of this may be summed up as follows -

**<u>The power of a computer combined with the simplicity of a phone.</u>**

Michael McLean B.A.(Wollongong) M. Litt.(Sydney)
MMC2.