1

The Palm OS Success Story

Since the release of the Pilot 1000 in 1996, devices running Palm OS have become synonymous with the term "handheld computer." Through the years, the designers of Palm OS have consistently been able to combine just the right mix of features to make a personal digital assistant (PDA) that is easy to integrate into almost any user's lifestyle. Designing an application that takes advantage of the strengths of the Palm OS platform requires an understanding of not only how the platform works but also why it was designed the way it was.

This chapter explains some of the thinking that has made the Palm OS platform so successful. It also points out important design considerations for developers of handheld applications. Finally, it provides an overview of the increasingly diverse world of Palm OS devices and the array of hardware capabilities they encompass.

The Palm OS Success Story

How has Palm OS maintained its position as the leader in handheld platforms, even in the face of capable challengers such as Microsoft Windows Mobile and Symbian OS?

One could debate the pros and cons of these and other worthy contenders for the title of "best handheld operating system" but the truth is that Palm OS continues to achieve a magic combination of simplicity and extensibility that attracts device manufacturers, developers, and users to the platform.

This success is all the more impressive when you consider that Palm OS has remained the leader throughout a period of time when the definition of a "PDA" has expanded from a simple personal organizer to a robust application platform — all the way through to the present time when it is becoming increasingly difficult to find a handheld that does not play music, take pictures, play games, or double as your cell phone.

What makes Palm OS a great platform for so many developers and handheld users? There are many reasons, but these are among the most important:

- **Palm OS is small, fast, and efficient.** Rather than suffer inevitable bloat as new features are added to the latest handhelds, Palm OS remains true to its core values and instead offers extensibility to its licensees, enabling them to build devices that add advanced features to the core operating system.
- **Palm OS is easy to use.** Devices that use Palm OS allow users to perform common tasks with a minimum of dialog boxes, menus, and screen navigation. Many common tasks are accomplished with a single button press or stylus tap.
- **Palm OS allows simple and fast desktop synchronization.** The Palm OS HotSync design enables one-button synchronization of data between the desktop and handheld. Despite having years to learn from Palm OS, other platforms have yet to approach the simplicity and ease of use that Palm OS HotSync offers.
- **Palm OS embraces diversity.** The number and diversity of licensees is a testament to how well the designers of Palm OS enable handheld manufacturers to adapt it to a wide variety of tasks, from multimedia to wireless communications. This is in direct contrast to other platforms, where devices tend to be fairly similar in form and function.

Palm hit upon a perfect combination of these factors with its first device, and it has resisted the temptation to cram marginally useful features into new Palm devices. Intelligent selection of such features has fashioned these devices into handy tools instead of merely expensive toys.

Comparing Desktop and Handheld Application Design

There are significant differences between a desktop computer and a handheld device — enough differences that designing a handheld application must be approached differently from designing a desktop application. Many elements must be kept in mind when designing a Palm OS application:

- Diversity of handheld form factors
- Expectation of performance
- Limited input methods
- Small screen size
- Processing power
- Battery life
- Limited memory
- RAM as permanent data storage

Diversity of Handheld Form Factors

Although certainly some desktop computers are more capable than others and often come with varying sets of peripherals, in general the vast majority of desktop computers are reasonably suited to run just about any desktop software application. By contrast, Palm OS handhelds are an extremely diverse target for the application developer to consider. The form factor and capabilities built into a given target handheld device may in fact determine whether or not your application makes sense for the target user, or indeed whether it will run at all.

Consider an application that depends on Internet connectivity. Whether or not a given target device supports a way to connect to the Internet is clearly going to dictate whether the device and its user will be a reasonable target for the application. What about smartphones? Are you willing to limit your application's audience by including functionality that depends on the presence of telephony features? How about if your application requires a high-resolution color screen? Is it worth it to create a low-resolution, grayscale version of your application for older devices?

Although this can be considered a challenge for the application developer, it is also a benefit. The designers of Palm OS have produced a unique platform that is adaptable to a wide range of handheld form factors. Handheld manufacturers have responded by producing Palm OS handhelds that, in many cases, are specifically oriented toward a certain type of user (for example GPS, telephony, or entertainment). As the developer, you can elect to take advantage of the knowledge that the owner of a given device is guaranteed to have a specific capability available to them.

Expectation of Performance

Desktop application users usually don't mind waiting a few seconds for a program to load because they plan to use the application for an extended period of time — and they probably aren't going anywhere anytime soon.

Compare this with a handheld user on the go. A person using a Palm OS handheld will need to look up a piece of data (such as a phone number) quickly or spend a few seconds jotting down a note, while in the middle of performing some other task. Someone who is talking to clients on the phone or trying to catch a bus doesn't have time to watch a "wait" cursor spin while an application loads.

Speed and efficiency are key to a successful Palm OS application. Writing fast code is only a small part of the equation; the user interface must be simple, intuitive, and quick to use. The application should allow for rapid selection and execution of commands. Functions that people use the most should require less interaction than those that are used less frequently.

Limited Input Methods

A desktop system is ideal for entering large quantities of data. A keyboard and a fast processor allow desktop users to input lots of text easily into the computer in a short period of time.

Modern Palm OS handhelds come with a variety of supported data input methods. Virtually all support the standard Graffiti method of entering special shorthand strokes using a stylus on the handheld screen. Graffiti works remarkably well for many users. Palm OS also supports a popup, onscreen keyboard. However, there are now several handhelds with tiny built-in QWERTY keyboards (such as the palmOne

Treo smartphone) and there are handhelds with a larger, laptop-sized keyboard (the Alphasmart Dana). There are also third-party add-on keyboards that communicate with the handheld by serial connector, IrDA, or Bluetooth.

With the possible exception of the Alphasmart Dana, and despite the numerous attempts made by handheld and smartphone manufacturers to make it easier to enter data on their devices, developers must realize that expecting a user to enter anything longer than a short note is asking a lot of the user.

As an alternative to direct input on the device, HotSync technology provides an easy way to get large amounts of data from the desktop to the handheld. One of the major advantages of Palm OS over other competing mobile platforms is the attention paid to making synchronization powerful yet easy to use. Many software applications leverage this capability by assuming that mass data entry will be performed on desktop machines, which then synch that data to their handheld. This kind of symbiosis between the desktop computer and the handheld plays to the strengths of both devices.

However, don't let this discourage you from writing applications that use a Palm OS handheld as a data collection tool. With intelligent interface design, you can perform data entry quickly and efficiently on such a device.

Small Screen Size

Current desktop machines have large monitors that generally run at a minimum resolution of 640 × 480 pixels, although with prices of display monitors continuing to decline, most computer users choose to run their systems at even higher resolutions. With this kind of screen real estate to play with, displaying large amounts of information and a complex user interface in the same space is easy.

By contrast, most Palm OS handhelds have a screen about six centimeters on a side, with a resolution of 160 × 160 pixels. Current high-resolution models support up to only 320 × 320 pixels of screen space, a far cry from the acreage available on a desktop computer. Even the new Cobalt version of Palm OS is designed to support a maximum of 320 × 480 pixels. Unlike desktop displays, keeping devices small enough for users to carry in their shirt pocket is a unique requirement for handhelds.

Designing applications to use such a small screen is a challenge. Displaying the right information is more important than fitting as much information on the screen as possible. You must strike a balance between showing enough information and keeping the interface uncluttered and simple to use.

Requiring users to scroll through several data screens to find the information they want will make your application frustrating to use. Find logical groupings of data and offer the user a way to filter different views of that data. The To Do List application is a good example of data filtering; its preferences allow users to choose quickly what subset of the list to display. Implementing the standard Palm OS user-defined categories also can help users zero in on exactly the data they want to view.

Unlike desktop machines, which are plugged into wall outlets and sport powerful and fast processors, Palm OS handhelds must rely on batteries for power, which limits them to slower processors. The small processor on such a device is not well suited to intense computations.

If your application has both handheld and desktop components, consider doing all the intensive number crunching on the desktop portion. A great example of relegating processor-intensive tasks to the desktop machine is Doc, the *de facto* standard for large text documents on Palm OS. Several converter applications exist for the desktop machine, which perform the computationally intensive conversion and compression

of a large text document to Doc format. The newly formatted document then can be transferred to the handheld during the next HotSync session. All that the Doc viewer application on the handheld needs to concern itself with is displaying the document; the faster desktop computer has handled all the hard stuff.

Processing Power

Most handheld models sold today sport ARM (Advanced RISC Machine) processors, which are much more powerful than the slower processors of the earlier Palm OS devices. Although developers must consider the fact that there are millions of older handhelds out there, some of the processing power limitations imposed on developers in the early days have been lifted, at least to a certain degree.

Graphics-intensive games and processor-intensive image converters and viewers are two examples of applications that only a few years ago would have been unthinkable. If you are faced with a similar computationally intensive task that is part of the application you are developing, you are still advised strongly to consider how important it is for that task to be performed on the handheld instead of being offloaded to the desktop. However, at least now you have an option.

Battery Life

Another factor related to processing power is the issue of battery life and your application's impact on it. Although a proliferation of devices can play music, video, and games and surf the Internet, these tasks take their toll on battery life. At a minimum, your application should not be such a drain on battery life that the handheld cannot survive a typical full day of usage without needing to be recharged.

Limited Memory

As memory prices continue to drop, desktop applications can afford to be less choosy about how they deal with memory. When your application has 64 MB or more to play with, it can load huge data structures into RAM and leave them there the entire time the program is running.

Compared to desktop computers, Palm OS handhelds have very limited memory space for running applications. Though Palm OS handhelds continue to grow in their total amount of RAM, with some topping out at 16 MB, only a small fraction of that (one-sixteenth or less) is available for dynamic memory allocation, application global variables, and static variables. Application stack space is even tighter, often only 4K or less. Versions of Palm OS prior to 3.0 have considerably less room, so designing applications that are compatible with older Palm OS handhelds can be somewhat challenging.

When designing your application, consider that such things as deeply recursive routines, large global variables, and huge dynamically allocated data structures are not Palm OS–friendly.

RAM as Permanent Data Storage

Hard drives provide desktop computers with abundant permanent storage for vast amounts of data. Palm OS handhelds have considerably more limited storage space because they must store both applications and data in RAM. Although many Palm OS handhelds support secondary storage on memory expansion cards, this storage is usually limited to data. Applications cannot run directly from an expansion card, so they must either reside in RAM or be copied to RAM from the card before running.

As of this writing, available memory on a reasonably modern Palm OS handheld ranges between 8 MB and 128 MB. This type of limited storage dictates that handheld applications remain as small as possible. Avoid adding features to your application that will be used infrequently; if a feature will be used by fewer than 20 percent of users, leave it out.

For example, features that globally modify an application's data, but will see only infrequent use, are prime candidates for inclusion in a companion program on the desktop. A command that removes duplicate entries in a database would be perfect for the desktop; it's not likely to be used very often on the handheld, and omitting it from the handheld application makes the program smaller and more efficient.

Your application should pack its data tightly before writing the data to memory. Not only will this reduce the amount of RAM required to store your application's data but it will decrease the amount of time that HotSync needs to synchronize the data with the desktop computer.

Designing Applications for Smartphones and Other Wireless Devices

From a communications perspective, the way in which a handheld or smartphone connects to other computers and peripherals definitely impacts how developers must think about features that they take for granted on desktop computers: accessing the Internet, local area networks, and peripherals such as printers. Considering the following:

- Many connection types
- Connection speed
- Mobile-user expectations
- Connection reliability

Many Connection Types

The primary method for a desktop computer to connect to the outside world is through a standard Ethernet jack. Although not as commonly as before, many computer users still connect to the Internet using a dialup modem. By contrast, smartphones and handhelds come equipped with many different connectivity options. Smartphones provide Internet access via wireless radios that support wide-area carrier data networks. Many handheld devices come with a built-in Bluetooth radio that enables short-range connectivity to other computers or even the Internet via a cell phone or access point connection. Some handhelds possess an 802.11b Wi-Fi radio that can connect over a somewhat larger distance to local area networks and the Internet from access points. Last but not least, every modern Palm OS handheld comes with infrared connectivity.

Consider an application that seeks connectivity to the Internet or other networked computers. Although the TCP/IP support in the Palm OS can insulate the application developer from needing to worry about the exact type of connection made, for short range connectivity TCP/IP may not be the most appropriate mechanism for the function in question. Furthermore, you must consider the ease with which the average user can make the connection required by your application.

Connection Speed

Desktop connectivity on today's networks generally occurs at 10 Mbps or greater. Fast Internet connection options afford the typical user connectivity to e-mail and the Web at bandwidths far in excess of 100Kbps.

Connection speed on smartphones and wireless handhelds varies greatly. Browsing the Web on a smartphone connection, often at speeds lower than that of a 56 Kbps dialup modem, can still be an exercise in frustration. Yet at the other end of the spectrum, multi-megabit connections can be achieved by Wi-Fi–enabled handhelds. Also to be considered are the upper limits of infrared (115 Kbps maximum) and Bluetooth (less than 1 Mbps).

Depending on what services and data transfer your application requires, the type of connection achievable by the user's handheld will significantly impact the end-user experience with your application.

Mobile-User Expectations

As we mentioned earlier in this chapter, a desktop user is comfortably seated in a chair and has the time and patience to wait for an application to deliver information. The structure of the Web, with its endlessly intricate links among pages of related information, is well-suited to the desktop user, allowing the user to browse the Web at a leisurely pace, for minutes or even hours at a time.

Mobile handheld users, on the other hand, are always in motion: standing, not sitting; on the go and in a hurry. Rather than leisurely browsing the Internet, handheld users who go online are doing so to find a quick answer to a question or to view a specific piece of information. If it takes more than a minute or two to find that information, odds are that the opportunity is lost and the user will simply become frustrated, give up, and move on to another task. Needless to say, the failure to deliver the information to the user in a timely fashion reflects poorly on the application as well as the device.

Accordingly, wireless applications need to be aware of a target average user who has an extremely short attention span. Needless to say, data access methods and screen navigation must be constructed so as to be as fast and direct as possible.

Connection Reliability

Glenn will never forget when he was a naïve young developer working on a piece of software that transferred data across a network between two personal computers. He had worked hard on his piece of the project and, when he was done, he proudly walked up to his boss and boasted that the work was complete and he could demo it for him. His boss walked over to the computer where his code was running and transferring data at a furious rate. He casually reached behind the computer and yanked the Ethernet cable away from the computer. With horror Glenn watched as his code died an ignominious death with its precious connection pulled out from beneath it. "Come back and let me know when you are really done," his boss said with a smile. Sheepishly, he walked back to his desk and began working on making his code much more bulletproof and robust in the face of a lost connection.

Today's desktop network connections seem fast and ubiquitous. The Ethernet cables snaking through our offices serve as a pleasant reminder that a connection is always there. Things are far different on smartphones and handhelds, where wireless, not wired, connections are the norm and the network can

(and often does) disappear at a moment's notice. Wireless applications need to be designed with the idea of an inherently unreliable connection in mind. Extended operations and protocols which rely on a lengthy, unbroken network session are generally not a good design. This is not to say that an application cannot be designed to download a 100K file. Rather, the application that does the downloading should consider the very likely possibility that the transfer will be interrupted by a break in the connection. How can your application plan for this possibility and reward, not frustrate, the end user?

Connecting to the Desktop

Sharing data with the desktop is a key ingredient in the popularity of Palm OS handhelds. It is one of the key competitive strengths of Palm OS versus other handheld platforms. The connection between the desktop and handheld allows each device to borrow the strengths of the other. A desktop computer is great for large-scale data entry and crunching numbers but not so great for carrying in your pocket when visiting clients. A handheld device is perfect for taking quick notes and reminding you of appointments but not so perfect for analyzing financial reports or writing a book. Together, both devices become greater than the sum of their parts.

The software component that forms the vital link between the Palm OS device and the desktop computer is the *conduit*. During synchronization with your handheld application, HotSync calls code in a conduit, which resides on the desktop computer. This code controls exactly what data HotSync transfers between the two devices. There are several different scenarios in which a conduit plays a vital role; the following are just a few examples:

- Two applications, one on the handheld and one on the desktop, use the conduit to keep records in their databases in synch with each other. This is how the conduit for the Date Book and the three other main Palm OS applications works. In this scenario, the conduit is responsible for looking at the records in both databases and determining which records are different between them, as well as in which direction the data must be transferred.
- The conduit keeps data in a handheld application synchronized with data in a centralized corporate database, either stored on the machine running HotSync or another machine on a corporate network. In this case, the conduit also might sift the data and transfer only a customized subset to the handheld based on user preferences. For example, a contact application for a sales force might download only information about businesses that a specific salesperson has been working with. Customization like this keeps the size of the data manageable and reduces the time required for HotSync to run.
- When syncing, the conduit compares content on the handheld with the contents of a Web page, mail server, or Usenet newsgroup. If the information on the server is newer than what the handheld application has stored, the conduit downloads the new data, processes it into a form that the handheld application can read, and then transfers it to the handheld. The conduit also may instruct the handheld application to discard out-of-date pages, messages, or articles. Because Internet connections are prone to delays, this sort of conduit should probably look only at information previously cached by a desktop application. A HotSync operation should be as short as possible because having the serial port open rapidly drains a Palm OS handheld's batteries.

If your application does not require the level of detailed synchronization logic that a conduit can provide, you may be able to use the default *backup conduit*. Instead of comparing the handheld application's

database record by record with the data on the desktop, the backup conduit simply makes a copy of the entire database and transfers it to the desktop computer. Although this works perfectly well for small application databases, it can slow down the HotSync process if your application stores a lot of data. For example, an address book database containing thousands of contacts may synchronize through a properly designed conduit in less time than it takes to copy the entire database to the desktop.

Chapter 21 provides an introduction to developing Palm OS conduits. Further details on writing conduits follow in Chapter 22.

Comparing Palm OS Handheld Devices

For earlier Palm OS developers, the main challenge in managing compatibility with the various handheld models out in the market was in maintaining support for devices running several earlier versions of Palm OS. Physically, the main variations to be found among these devices were the amount of memory and the availability of an expansion card slot. Except for applications that sought specifically to work with expansion cards, well-written application code could be expected, generally speaking, to function across the range of devices available on the market.

Since 2003, the proliferation of form factors and hardware features in devices has been remarkable. The variations among devices today are far greater than just a few years ago. You can categorize these variations as follows:

- Multimedia (audio/video)
- Screen size and orientation
- Voice/telephony
- Imaging/photo capture
- Data input/keyboard
- Navigation/scrolling
- Wireless data connectivity

As a result of these variations, developers are less focused on the basic device specs (amount of memory, processor speed) and much more concerned instead on the type of user who will buy a given device, and the types of applications and functionality that user will be interested in.

Complicating matters is the fact that most devices on the market today combine several of these variations into one device. For example, the palmOne Tungsten C has wireless data, navigation, and a built-in keyboard. The Alphasmart Dana has an extended screen size, keyboard, and wireless data. The popular palmOne Treo 600 smartphone has managed to cram in just about all of these categories: It's a phone, it can play music, it can take pictures, and it has advanced navigation and data entry features.

There are those who have long sought after the elusive one perfect device that replaces their phone, PDA, personal audio player, and so on. The reality is that although there will be attempts to produce such a device, they will inevitably involve compromises in one or more areas of functionality. Most

devices, instead, will continue to focus on being excellent at the kinds of things that the devices' target users want to do with them. This doesn't make one device better than another; it just makes it different.

Even in a time when most people are predicting that the majority of handhelds in the future will be smartphones, and voice capability will be commonplace in PDAs, the mix of other capabilities included on these smartphones will continue to be fine-tuned to the needs of the target user. As a case in point, the ubiquity of the expansion card slot — a feature that was brand new just a few years ago — did not result in Palm OS devices' becoming boring and homogeneous. Rather, it simply graduated from being a bonus feature in high-end handhelds to becoming a standard part of every handheld.

Looking to the Future

The official Palm OS documentation stresses the importance of developers not making assumptions about the hardware that underlies their applications. This is important because the hardware may change. If your code ignores the Palm OS APIs and directly accesses the hardware, your application is very likely to break on future devices. Furthermore, as we've detailed in this chapter, the diversity of Palm OS devices virtually guarantees that your application will encounter different hardware options and configurations from one handheld to the next.

PalmSource, the company that publishes Palm OS, continues to meet the challenge of extending the platform to embrace new and wonderful handheld capabilities, while at the same time minimizing the impact of these changes on developers and their applications. The divergence of Palm OS into the Garnet and Cobalt versions is a perfect example of this kind of tradeoff: offering developers the assurance that their investment in Palm OS code will be preserved, while offering new APIs and functionality to new devices and applications.

> *Use the Palm OS APIs instead of making direct calls to hardware. The Palm OS APIs are very complete. If you stick with using the provided functions, your application should continue to run smoothly on new devices.*

Summary

This chapter explained the philosophy behind the Palm OS platform and introduced you to the unique mindset required to design effective handheld applications. You should now know the following:

- The Palm OS platform's success is based upon a compact and efficient operating system, ease of use, excellent desktop integration, and support for diverse handheld form factors.
- Handheld application development is very different from desktop application development because it requires you to work within a number of constraints.
- Applications that run on smartphones and wireless devices can expect to have different connectivity options, user expectations, and connection reliability from applications running on desktop systems.
- The diversity of Palm OS handhelds is on the increase. A variety of different hardware features is becoming available in various combinations oriented toward specific user types.