Software Development for the Palm OS

Jan Schaumann

06/08/00

Abstract

A brief overview of Software Development for the Palm OS, Webclipping, Conduits, compiled from various websites.
## Contents

1 Why Programming for the Palm OS is different 3  
   1.1 Screen Size ................................................. 3  
   1.2 Quick Turnaround Expected ................................. 3  
   1.3 PC Connectivity ............................................. 3  
   1.4 Input Methods ............................................... 3  
   1.5 Power ....................................................... 4  
   1.6 Memory ..................................................... 4  
   1.7 File System .................................................. 4  
   1.8 Backward Compatibility ..................................... 4  

2 Palm OS Programming Concepts 4  
   2.1 PilotMain ..................................................... 5  
   2.2 Event Loop .................................................. 5  
   2.3 User Interface ............................................... 5  
   2.4 Memory ..................................................... 5  
   2.5 Function calls ............................................... 5  

3 Tools 5  
   3.1 Coding Tools ............................................... 5  
   3.2 Emulators ................................................... 6  
      3.2.1 Obtaining ROM Image Files ............................. 6  
      3.2.2 The Palm OS Emulator ................................. 6  
      3.2.3 Other Palm OS Emulators ............................... 7  
   3.3 Conduit Development Kits (CDKs) ......................... 7  
   3.4 Web Clipping Development (WCA) ........................ 8  
      3.4.1 Two components to WCA ............................... 8  
      3.4.2 WCA Builder Tools ..................................... 9  
      3.4.3 A WCA Example ....................................... 9  
      3.4.4 Palm Query Applications (PQA) ....................... 10  
      3.4.5 A PQA Example ....................................... 11  

4 Hardware using the Palm OS 12  

5 Wishlist 12  

6 Main Resources 13
1 Why Programming for the Palm OS is different

Developing applications for the Palm OS is a bit different than developing applications for desktop computers such as the PC or a Mac. First off, the Palm computing platform is designed differently than a desktop computer. Also, users simply interact with the device differently than they do desktop computers.

1.1 Screen Size

The Palm OS device’s screen is only 160x160 pixels, so the amount of information you can display at one time is limited. For this reason, the user interface must be designed carefully with different priorities and goals than are used for large screens. Note that screen sizes of future Palm OS devices may vary.

1.2 Quick Turnaround Expected

On a PC, users don’t mind waiting a few seconds while an application loads because they plan to use the application for an extended amount of time. By contrast, the average Palm user uses a Palm application 15 to 20 times per day for much brief periods of time, usually just a few seconds. Speed is therefore a critical design objective for hand-held organizers and is not limited to execution speed of the code. The total time needed to navigate, select, and execute commands can have a big impact on overall efficiency. (Also consider that the Palm OS does not provide a wait cursor.)

To maximize performance, the user interface should minimize navigation between windows, opening of dialog boxes, and so on. The layout of application screens needs to be simple so that the user can pick up the product and use it effectively after a short time. It’s especially helpful if the user interface of your application is consistent with other applications on the device so users work with familiar patterns.

1.3 PC Connectivity

PC connectivity is an integral component of the Palm Computing platform device. The device comes with a cradle that connects to a desktop PC and with software for the PC that provides "one-button" backup and synchronization of all data on the device with the user’s PC.

Many Palm OS applications have a corresponding application on the desktop. To share data between the device’s application and the desktop’s application, you must write a conduit. A conduit is a plug-in to the HotSync technology that runs when you press the HotSync button. A conduit synchronizes data between the application on the desktop and the application on the hand-held device. To write a conduit, you use the Conduit SDK, which provides its own documentation.

1.4 Input Methods

Handheld users don’t have a keyboard or mouse. Users enter data into the device using a pen. They can either write Graffití strokes or use the keyboard dialog provided on
the device.

While Graffiti strokes and the keyboard dialog are not as convenient as using the full-sized desktop computer with its keyboard and mouse. Therefore, you should not require users to enter a lot of data on the device itself.

1.5 Power

The Palm Computing platform device runs on batteries and thus does not have the same processing power as a desktop PC. You should keep this in mind when developing applications for the Palm OS and avoid using heavy calculation.

1.6 Memory

The Palm OS device has limited heap space and storage space. Different versions of the device have between 512K and 8MB total of dynamic memory and storage available. The device does not have a disk drive or PCMCIA support. Because of the limited space and power, optimization is critical. To make your application as fast and efficient as possible, optimize for heap space first, speed second, code size third.

1.7 File System

Because of the limited storage space, and to make synchronization with the desktop computer more efficient, Palm OS does not use a traditional file system. You store data in memory chunks called records, which are grouped into databases. A database is analogous to a file. The difference is that data is broken down into multiple records instead of being stored in one contiguous chunk. To save space, you edit a database in place in memory instead of creating it in RAM and then writing it out to storage.

1.8 Backward Compatibility

Different versions of the Palm Computing platform device are available, and each runs a different version of the Palm OS. Users are not expected to upgrade their versions of the Palm OS as rapidly as they would an operating system on a desktop computer. Updates to the OS are designed in such a way that you can easily maintain backward compatibility with previous versions of the OS, and thus, your application is available to more users.

2 Palm OS Programming Concepts

Palm OS applications are generally single-threaded, event-driven programs. Only one program runs at a time. To successfully build a Palm OS application, you have to understand how the system itself is structured and how to structure your application.
2.1 PilotMain
Each application has a PilotMain function that is equivalent to main in C programs. To launch an application, the system calls PilotMain and sends it a launch code. The launch code may specify that the application is to become active and display its user interface (called a normal launch), or it may specify that the application should simply perform a small task and exit without displaying its user interface. The sole purpose of the PilotMain function is to receive launch codes and respond to them.

2.2 Event Loop
Palm OS is an event-based operating system, so Palm OS applications contain an event loop; however, this event loop is only started in response to the normal launch. Your application may perform work outside the event loop in response to other launch codes.

2.3 User Interface
Most Palm OS applications contain a user interface made up of forms, which are analogous to windows in a desktop application. The user interface may contain both predefined UI elements (sometimes referred to as UI objects), and custom UI elements.

2.4 Memory
All applications should use the memory and data management facilities provided by the system.

2.5 Function calls
You implement an application’s features by calling Palm OS functions. Palm OS consists of several managers, which are groups of functions that work together to implement a feature. As a rule, all functions that belong to one manager use the same prefix and work together to implement a certain aspect of functionality.

3 Tools
3.1 Coding Tools
Other development (commercial) tools for the Windows platform are:

- **CASL (Compact Application Solution Language)**: Compact Application Solution Language Tools from Feras Information Technologies is a Windows-based integrated development environment for creating high-level interpreted applications integrated with conduits to Windows desktop applications. Includes a development environment, a conduit, and sample code.
- **Satellite Forms**: Satellite Forms from Puma Technology is a visual, rapid development environment for applications and conduits. Design applications via drag-and-drop. Develop event-driven scripts in a language similar to Visual Basic using custom controls, plug-ins, and Satellite Forms APIs.

- **Pendragon Forms**: Pendragon Forms from Pendragon Software is a fast and simple way to create data collection applications and conduits.

Since Applications for the Palm OS are written in C/C++, you write the code in any environment and then compile it. You do need the Palm OS SDK, which includes the headers, libraries, and tools for Palm Computing platform development on Windows, Mac OS, or Unix/Linux platforms. In order to compile the source code, you need a couple of tools, which are (for Unix/Linux and Windows at least) freely available.

### 3.2 Emulators

The Palm OS Emulator is software that emulates the hardware of the various models of Palm OS platform devices. It is extremely valuable for writing, testing and debugging applications. Create “virtual” handhelds by running the Emulator on Windows, Mac OS, or Unix/Linux computers.

Emulated devices can be loaded with new ROMs, so you can test your application with virtual devices, including different devices than you own, foreign language ROMs, and debug-enabled ROMs for extra error checking and debugging features.

The Emulator software does not include ROM images. It is like a computer without an operating system. There are two sources of ROM images: ROM images file or upload a ROM from an actual device. As a developer, there are advantages to using ROM image files because you can use “debug-enabled” ROMs that contain extra debugging code not found in shipping hardware ROMs.

#### 3.2.1 Obtaining ROM Image Files

For legal reasons, you must sign an agreement with Palm Inc. to download ROM images. To obtain the Prototype License and Confidentiality Agreement, you must be a member of the Palm[tm] Solution Provider Program. After joining the Solution Provider Program, we recommend that you join the Developer Seeding Program, which requires that you return a signed agreement to us. This allows you to access ROM images and other valuable prerelease information. If you are based in the USA, you may also obtain images using a "clickwrap" agreement that does not require a signed agreement.

You can also download a ROM using the Emulator from a handheld device that has been placed into the cradle connected to your desktop computer.

#### 3.2.2 The Palm OS Emulator

The Palm OS Emulator (or POSE, former “copilot”) displays an on-screen image that looks exactly like a Palm connected Organzier. You can select which kind of Palm
handheld device you want to emulate, and you can also specify that you want POSE to display the screen in double size, which continues to provide an accurate presentation and makes the Palm screen easier to view.

You can use your mouse on your desktop computer just as you use a stylus on a Palm Device. You can even use the Grafiti Power Writing Software with POSE and your mouse. You can use POSE to perform some debugging of your applications and you can use the Emulator with Palm Debugger to debug in exactly the same manner as with your application running in the actual hardware handheld device.

3.2.3 Other Palm OS Emulators
The Palm OS Emulator was based on "Copilot", written by several different developers, and is available on the following platforms:

- Copilot for Mac OS (PowerPC) - http://members.aol.com/illumesoft/copilot.html
- “Zilot” for Mac OS - http://www.pollet.net
- “xcopilot” for Unix/Linux - http://www.isaac.cs.berkeley.edu/pilot/
- “Copilot/2” - URL currently unavailable
- “PilotCE” for WindowsCE - http://www.geocities.com/SiliconValley/Monitor/8604/Beta/PilotCE/

3.3 Conduit Development Kits (CDKs)
Conduit Development Kits (CDKs) are tools to develop conduits. Conduits are software plug-ins for the HotSync Manager application. They exchange and synchronize data between a desktop computer and a Palm OS platform handheld computer.

Most conduits synchronize data such that data on the handheld mirrors the data on desktop. Conduits also transfer, import/export data, or cause Palm OS applications to be installed. Generic conduits are Palm Inc.’s future direction for conduits. The idea is to provide an API for conduit development to be compiled for Windows and/or Mac OS. The details for making it work on each platform will be hidden within the API.
The CDKs are available for the following platforms:

- Conduit Development Kit - Windows C/C++ Edition (uses Microsoft Visual C++ 6.0)
- Conduit Development Kit - Windows Java Edition (for developing Palm OS Applications in Java - uses Symantec Visual Café Pro for Java)
- Conduit Development Kit - Mac OS Edition (uses Metrowerks CodeWarrior Pro for Mac OS)

All these CDK’s are available to download for free from http://www.palmos.com/dev/tech/conduits/

3.4 Web Clipping Development (WCA)

The web clipping architecture includes client-side applications that run on a Palm OS device, proxy servers for handling translation between the web clipping application format and HTML, and content servers. The client-side application is called a web clipping application. It is constructed in HTML and translated into the web clipping application format. Once this application is installed on a Palm OS platform device, content is delivered from the content provider’s own web site as a subset of the HTML 3.2 standard.

If you do not have access to hardware that supports web clipping, you can use software to emulate the hardware with the POSE. When emulating a web clipping enabled device, you use your desktop computer network to emulate wireless access free of charge. Web clipping applications use a special proxy to get web pages and do not connect directly to web sites.

Web clipping applications are written in a subset of the HTML 3.2 specification. Many of the usual tags for things like tables, forms, checkboxes, radio buttons, select lists, and font sizes are supported. However, in order to work well even though constrained by the limited bandwidth available on the wireless connection, web clipping applications do not support more resource intensive features such as JavaScript, nested tables, frames, cookies, or Java.

Web clipping applications are designed to be small and to give users the information they need with minimum effort and maximum speed. Likewise, results coming from your server to a handheld device are designed to be small and to the point. Remember, this is not web browsing, but web clipping. Web clipping applications should be written to provide users with only the information that they need without any extraneous items. Try to limit the size of your result pages to about 400 bytes.

3.4.1 Two components to WCA

There are two main components to writing successful web clipping applications: the web clipping application that you install on the device, and the backend server that returns results pages to the device.
• **Web clipping application** - This is an application that you build using HTML, which is installed by the end user onto their handheld organizer. A web clipping application is like a mini-web site that is stored locally, so access to it is instant and free. The index page of the web clipping application usually contains either a form or a list of links, which are the gateways to the live data provided by your server. Static information, such as graphics, help, and "about" pages should also be placed within a web clipping application.

• **Results pages (clippings)** - These pages are returned by your backend server after they receive a request from your web clipping application. They are written in HTML and will often link to images and HTML documents that are stored in the web clipping application so that this static information doesn’t need to be transmitted over the air.

### 3.4.2 WCA Builder Tools

The WCA Builder Tools needed to build WCA’a (ie translate the HTML-like code into “Palmspeak”) are available to download for free from [http://www.palmos.com/dev/tech/webclipping/](http://www.palmos.com/dev/tech/webclipping/).

### 3.4.3 A WCA Example

**The Code:**

```html
<html>
<head>
    <title>Used Macs</title>
    <meta name="palmcomputingplatform" content="true">
</head>
<body>
    <center><img src="macos.gif"></center>
    <h2 align=center>Used Mac Prices</h2>
    <p align=center>Search for the most up-to-date prices on Macintosh computers:</p>
    <center>
        <form method="post" action="http://www.myserv.com/cgi-bin/search.pl">
            <select name="model">
                <option>Mac 128K</option>
                <option>Mac 512K</option>
                <option>Mac Plus</option>
                <option>Mac SE</option>
                <option>Mac SE/30</option>
                <option>PowerMac 7100</option>
                <option>PowerMac 8100</option>
            </select>
            <br><br><br>
            <input type="submit" value="Search" name="go">
        </form>
    </center>
</body>
</html>
```
We see that WCA's consist almost entirely of (very simple) HTML. The only thing that differs from pure HTML is the meta-tag `<meta name="palmcomputingplatform" content="true">`. Images, if used, are stored locally on the device, since it is not recommended to transfer images over the wireless connection (bandwidth!) - the WCA Builder will enter them into the appropriate database on the device.

After passing this code through the WCA Builder, the application would look something like this:

### 3.4.4 Palm Query Applications (PQA)

Web clipping applications are Palm record databases that are read by the Web Clipper application (Web Clipper) on the Palm VII[tm] handheld (or other device that supports web clippings). Pages and images are stored as separate records in the database, and the number of records listed in the Records view of the Applications Launcher Info screen
correspond to the total number of pages and images that comprise the web clipping application.

With a couple of function calls you can enhance your WCA's easily with your C++ programs, and vice versa (ie enhance your C++ Application with WCA's). You can:

- create “pseudo-cgi’s” on your Palm Device (e.g. parsing input from a URL)
- process data from web clipping pages (e.g. graphing, updating databases etc.)
- store data on the handheld for WCA's (e.g. preferences)

3.4.5 A PQA Example

The Code:

```html
<html>
<head>
<title>"palm" Test</title>
<meta name="PalmComputingPlatform" content="true">
</head>
<body>
<h1 align="center">Special Function Palm Tester</h1>
<p><a href="palm:memo.appl">Memo Pad</a>
</body>
</html>
```

After passing this code through the PQA Builder, the application would look something like this:
4 Hardware using the Palm OS

There are several types of hardware projects for the Palm OS® Platform:

- **Palm Inc.** - http://www.palmos.com/dev/tech/hardware/
- **Symbol Technology Development** -
- **TRG** - http://www.trgnet.com/
- **Qualcomm (Smartphone)** - http://www.palmos.com/dev/tech/qualcomm/

5 Wishlist

To guarantee maximum efficiency every person involved in the actual coding process should be able to work in an environment (s)he is most comfortable with. While some
people may prefer coding, compiling and debugging “by hand”, others may prefer a
complete Integrated Development Environment (IDE). The most widely used tool
in Palm OS Software Development is the CodeWarrior Interactive Development En-
vironment (IDE) from 3Com Corporation. Documentation for the CodeWarrior IDE
is provided with CodeWarrior. CodeWarrior is available for Windows, Macintosh,
Solaris, Linux and Java. CodeWarrior is proprietary software, currently priced ≈$90.
Even though almost all information and documentation one might need to develop
applications for the Palm OS is available on the Internet, many people prefer having
a decent book (yes, paper!) as desktop reference handy. The one book most suitable
for this purpose seems to be O’Reilly’s “Palm Programming: The Developers Guide”
(http://www.oreilly.com/catalog/palmprog/).
Needless to say that everybody involved should have PalmPilot or other Palm OS
based handheld device to familiarize themselves with the environment we develop for.

6 Main Resources

Programming Palm OS in a Nutshell -
http://www.palmos.com/dev/tech/docs/palmos/Nutshell.html