Web Clipping Developer's Guide

CONTRIBUTORS

Written by Gary Hillerson
Production by `<dot>PS` Document Production Services
Engineering contributions by Kathy Ly, Scot Stennis, Michel Turcotte

Web Clipping Developer's Guide
Document Number 3009-003
May 1, 2001
For the latest version of this document, visit
http://www.palmos.com/dev/tech/docs/.

Palm, Inc.
5470 Great America Pkwy.
Santa Clara, CA 95052
USA
www.palmos.com

Table of Contents

1

Introduction

The *Web Clipping Developer's Guide* shows you how to develop web content for the Palm OS® platform. Creating such content involves the design and development of two different components:

- The web clipping application that resides on Palm Powered™ handhelds. You create this application on your personal computer, build it with the Web Clipping Application Builder tool, and then install it on the handheld device. The user taps on this application to start interacting with the Internet.
- The content that is received from the Internet in a form called web clippings, which are displayed on the handheld screen.

As a developer, you create the content. Users can download your web clipping application and use it as a gateway to the web clipping content that is managed on a server on the Internet.

This book provides you with the information you need to create and customize both components, including important style and design guidelines that address the unique requirements of handheld web users.

NOTE: This book uses the term web clipping devices to refer to the subset of Palm Powered handhelds that support web clipping capabilities. Some Palm devices support web clipping directly, and others can be upgraded with the Palm Mobile Internet Kit (MIK). For more information, you can review the hardware comparison page on the Palm Developer web site, which is located at the following URL: http://www.palmos.com/dev/tech/hardware/compare.html.

This chapter provides an overview of the web clipping architecture, including an introduction to web clipping applications and web clippings.

Chapter 2, "Getting Started with Web Clippings," provides you with a quick start to developing a fully operational web clipping application that can send queries to and display results from a web site on the Internet. You can follow the step-by-step instructions in this chapter to quickly learn how to develop and test a web clipping application.

Chapter 3, "Web Clipping Style Guidelines," discusses standards and guidelines for designing the user interface to web clippings.

Chapter 4, "Creating Web Clipping Applications," describes how to perform specific tasks in your web clipping applications, such as using images, using forms, and creating multi-platform HTML. This chapter builds on the information presented in Chapter 2.

Chapter 5, "Advanced Web Clipping Actions," adds information for advanced developers who wish to include more sophisticated capabilities in their web clipping applications, such as launching a Palm OS application, managing cookies, or interfacing with the INetLib code library. This chapter builds on the information presented in Chapter 4.

Chapter 6, "HTML for Web Clipping Applications," is the reference guide for HTML use in your web clipping applications. It includes descriptions of all of the meta tags and other extensions that Palm, Inc. provides.

Chapter 7, "Using the Web Clipping Application Builder," describes the Web Clipping Application Builder tool, which you use to convert your HTML and image files into an executable web clipping application that runs on Palm Powered handhelds.

Appendix A, "Web Clipping Transaction Errors," describes the common errors that you might see in a web clipping application.

Appendix B, "Palm HTML Summary," provides a summary description of the HTML that you can use in web clippings.

Appendix C, "Palm OS HTML 3.2 DTD," is the official Document Type Definition (DTD) understood by the Palm OS HTML engine.

Appendix D, "Web Clipping Port Assignments," describes the HTTP and HTTPS port assignments you can use with web clippings.

Appendix E, "Palm OS Lz77 Compression Details," provides detailed information about the Lz77 compression scheme used with web clippings.

Overview of Web Clipping Architecture

Palm, Inc. invented web clippings to make it possible for users to easily access information on the Internet with a small screen and low connection bandwidth. Web clipping technology allows users to extract and receive specific information from a web page, much like clipping a specific article out of a newspaper.

Numerous web sites are now enabled for web clipping, which means that the site's content is available in web clipping format. In the typical scenario, a web clipping application running on a Palm Powered handheld sends a query to the web site. The web site responds to the query by sending a clipping back to the handheld, and the web clipping application displays the returned clipping.

Figure 1.1 shows an overview of the web clipping architecture. The highlights of the architecture are as follows:

- You create a web clipping application by compiling standard HTML 3.2 pages with the Web Clipping Application Builder tool. This program generates a `.pqa` file that operates like a mini-web site and can execute on Palm Powered handhelds.

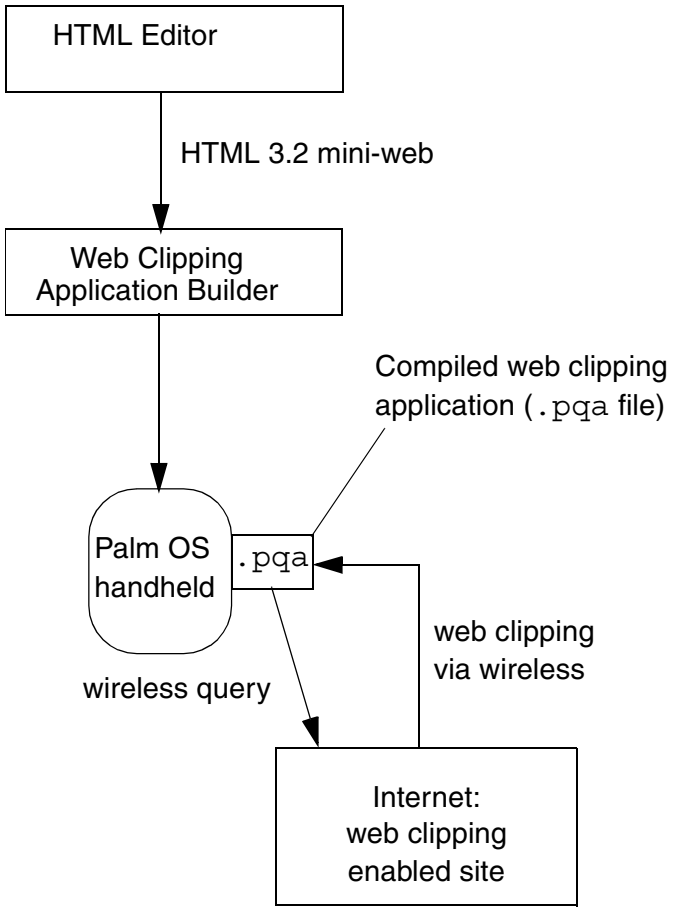
  A `.pqa` file is installed on Palm Powered handhelds as a Palm records database that contains all of your web clipping application content. The Web Clipping Application Viewer (the Viewer) opens and displays `.pqa` databases.

- When the user invokes your web clipping application, the Viewer opens your web clipping application and renders its contents. The Viewer is the Palm OS application that displays web clipping applications and handles communications with the Palm proxy servers. For more information about the proxy servers, see About the Palm Web Clipping Proxy Servers.

NOTE: The Web Clipping Application Viewer is sometimes referred to as the Web Clipper or as the Clipper.

- The web clipping application sends a query to the Internet.
- A server-side page or program (for example, an application or CGI script) responds to the query by generating an HTML clipping to send back to the web clipping application.
- The Viewer displays the query results (the clipping) on the handheld screen.

Figure 1.1 Overview of web clipping architecture



About Web Clipping Applications

A typical web clipping application contains an HTML form that displays information and allows the sending or requesting of information, hyperlinks to other information. The information can

be stored locally, on the handheld device, or remotely, on a web site that can be accessed on the Internet. A web clipping application can also be a self-contained web site.

NOTE: If you are a new web clipping application developer, the quickest and easiest way to get started is to follow the steps in Chapter 2, "Getting Started with Web Clippings," and create your first web clipping application.

Figure 1.2 shows a typical web clipping application. This application contains a number of links; when the user taps on any of these links, the HTTP(S) request is sent to the Internet, and the web site sends back a web clipping page that is displayed on the handheld screen.

Figure 1.2 Typical web clipping application

Creating a Web Clipping Application

You need to complete two steps to create a web clipping application:

1. First, use an editor on your personal computer to create the HTML content for display on the Palm Powered handheld. You can create a single HTML page or a web of related pages.

   The main page of most web clipping applications is a form or a list of links. The user interacts with the form or list and sees results on a new page or on the same page. For more information, see Chapter 4, "Creating Web Clipping Applications."

2. Next, compile the HTML content with the Web Clipping Application Builder tool. This program creates a web clipping application, with `.pqa` extension, that the user can transfer to a Palm Powered handheld. When the user runs the `.pqa`, the Viewer opens and displays the web. See Chapter 7, "Using the Web Clipping Application Builder" for more information about building your web clipping applications.

NOTE: Although `.pqa` files look like applications to users, they are actually special database files opened and displayed by the Web Clipping Application Viewer software in the Palm OS.

By default, when a user installs your compiled web clipping application on his or her Palm Powered handheld, the Applications Launcher shows the application in the Palm.Net category. Your web clipping application can be moved to any category, just like any other Palm OS application. The user taps the .pqa icon to display your web clipping application. Figure 1.3 shows the Palm.Net category in the Application Launcher of a typical Palm Powered handheld.

Figure 1.3 Palm.Net category in the Application Launcher

Additional Capabilities

In addition to interfacing with web sites, your web clipping application can use advanced features that Palm provides, including the ability to:

- launch or sublaunch another Palm OS application

- send mail
- upload and download binary files
- determine device location and capability information at runtime

Chapter 4, "Creating Web Clipping Applications," describes how to use these and other features in web clippings.

Chapter 6, "HTML for Web Clipping Applications," provides reference information for the Palm extensions to HTML for use in web clippings.

If you are a Palm OS developer, you can also directly access the Internet by calling INetLib functions, as described in the *Palm OS Programmer's Companion*, vol. II, *Communications* and *Palm OS Programmer's API Reference* books.

About Web Clippings

Web clipping applications display web clippings that are sent back from web sites. Most web clippings are small HTML pages that are dynamically generated on a server by a program, which can be an application or something else, such as a CGI script. A web clipping can also be a small, static HTML page. Figure 1.4 shows a typical web clipping that was sent back in response to tapping a link in a web clipping application.

Figure 1.4 A typical web clipping

A web clipping is distinguished from a standard web page in that a clipping is designed specifically to work with two restrictions that apply to Palm Powered handhelds:

- wireless devices such as the Palm VII™ have small screens
- currently, the most common wireless interfaces to the Internet are slow compared to dial-up access from a home computer

This means that most clippings strive to reduce the amount of data that is sent from the Internet to the handheld device. Web clipping designers typically use the following approaches to reduce the amount of data that needs to be transmitted:

- Structure the information to display what the user wants to see on one page.
- Limit font changes and other attributes that increase the size of the HTML file that is sent to the Viewer.
- Reduce the use and size of graphic images.
- Whenever possible, include static images in the `.pqa` file so they don't have to be transmitted.
- Reduce the width of tables so they can fit on the handheld screen without horizontal scrolling.

Chapter 3, "Web Clipping Style Guidelines," includes additional information about designing web clippings for an improved user experience.

NOTE: The Viewer will display standard (non-web clipping) web pages; however, these pages might not display as expected, and the user cannot control how they do display. Palm, Inc. strongly encourages you to create customized web clipping pages for display on Palm Powered handhelds.

Enabling Web Clippings on a Web Site

Enabling web clippings on a web site means that the site can generate "Palm friendly" web pages (clippings) in response to a query from a web clipping application. Palm friendly pages are standard HTML 3.2 pages that have been customized for transmission to and display on Palm Powered handhelds.

Many web sites generate web clippings dynamically with a CGI script. One important quality of Palm friendly web pages is that

each page contains the `PALMCOMPUTINGPLATFORM` meta tag, which tells the proxy servers that the page has been customized as a web clipping.

Behind the Scenes of Web Clipping

Figure 1.5 shows an overview of how web clipping works. To summarize:

1. The user launches a web clipping application from the Application Launcher.
2. The Viewer activates and displays the top-level page (typically named `index.html`) of the web clipping application.
3. When the user clicks on a link or submits a form in the web clipping application, the Viewer compresses the query into a Palm Query Format packet and sends the packet to the local base station.

   If the query uses the HTTPS protocol, the request data is encrypted using Certicom's elliptic curve cryptography (ECC) technology before being sent.
4. The local base station relays the compressed packet to the Palm web clipping proxy servers at the Palm.Net data center.
5. The proxy server that receives the packet converts it into a standard packet and sends that packet to the destination server on the Internet.

   If the request is encrypted, the proxy server first decrypts it, and then encrypts it using a 128-bit key. The server then establishes a SSL connection with the destination content server and sends the request using the HTTPS protocol.

   If the request is not encrypted, the proxy server sends a standard HTTP request to the destination content server.
6. When the proxy server receives information back from the Internet, it performs the complementary process, converting

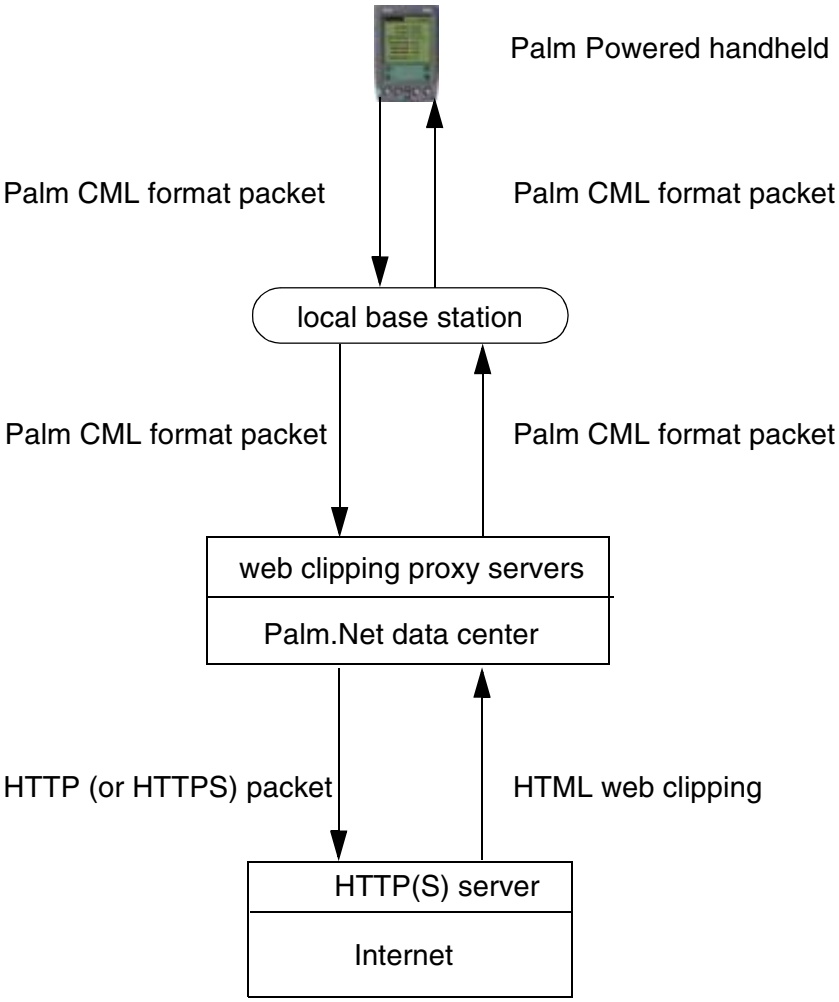
that information into a (compressed) Palm Proxy Format packet, which it then sends to the local base station.

If the original request was encrypted, then the proxy server uses Certicom elliptic curve cryptography to encrypt the compressed web clipping.

7. The local base station relays the packet to the handheld device, where the Viewer displays the query result. If the packet is encrypted, the Viewer decrypts it before rendering the clipping on the screen.

Chapter 6, "HTML for Web Clipping Applications," provides a more complete description of how web clipping works in conjunction with the Palm proxy servers to provide Internet access for Palm Powered handhelds.

Figure 1.5 Overview of web clipping operation



About the Palm Web Clipping Proxy Servers

The Palm proxy servers provide an interface between web clipping applications and the Internet. When a web clipping application requests information from the Internet, the request is sent to a proxy server that performs a transformation on the request and then sends it to the Internet. And when a web site sends a page back to the Palm Powered handheld, it is sent to a proxy server, which transforms the page and then sends it on to the handheld device.

Palm uses proxy servers to facilitate the compression of information that is sent to and from Palm Powered handhelds. This increases the

efficiency of accessing the Internet, improves security, and allows the proxy servers to filter out any HTML in the response that the Viewer does not support.

To understand more about the use of the Palm proxy servers, you need to know a little bit about how the Viewer renders HTML pages on Palm Powered handhelds. Although you write your web clipping applications in HTML, the Web Clipping Application Builder (WCA Builder) program saves your executable web clipping applications in a compressed format known as Compressed Markup Language (CML). This is the format that the Viewer renders on Palm screens.

When a web site sends HTML back to a handheld device, the proxy server converts the HTML into CML and then sends it on. In addition to compressing the web page for increased display speed, the proxy server:

- Strips out any content that is enclosed between the `<smallScreenIgnore>` and `</smallScreenIgnore>` tags. Web page developers can use these tags to include content that is displayed on desktop computers and is ignored on smaller screens.
- Strips out certain HTML content that the Viewer does not handle, including cascading style sheets, nested tables, and Java applet code. For a complete list, see Unsupported HTML Features in Chapter 6, "HTML for Web Clipping Applications."
- Transforms images for transmission to the Palm Powered handheld.

New Feature

If the handheld is running version 4.0 or later of the Palm OS, then device information that includes the bit-depth capability of the device, is automatically sent to the proxy server. The proxy server can then modify the bit depth of images that it sends to the device. For example, if a web page includes a 16-bit color image and the device supports 2-bit gray scale, then the proxy server sends a 2 bit deep version of the image,

which can significantly reduce the amount of data that has to be sent to the device.

If the handheld is running an earlier version of the Palm OS, then the proxy server sends a 2 bit deep version of the image, since earlier versions of the Viewer could only display those kinds of images.

NOTE: Device capability information is automatically sent to the proxy servers; however, this information is not automatically sent to the content server on the Internet. If you want to explicitly send device capability information to the Internet and you're using Palm OS 4.0 or later, you can use the `%WCDevCaps` variable, which is described in The %WCDevCaps Variable in Chapter 6, "HTML for Web Clipping Applications."

Palm Powered handhelds that support web clipping interface with the proxy server farms managed by the service to which the user subscribes. For example, the Palm.Net proxy servers are used for Palm VII and Mobile Internet Kit devices, and the OmniSky proxy servers are used for Palm V™ series devices using a modem branded for OmniSky.

If you are using the Palm OS Emulator (POSE) program, you can interface instead with one of two proxy servers that Palm has made accessible. You need to establish an Internet connection from your desktop computer, and configure POSE to interface to the proxy server, as described in 4. Configure POSE for Web Clippings in Chapter 2, "Getting Started with Web Clippings."

Caching of Clippings

The Palm web clipping proxy servers do not cache clippings. However, some web clipping proxy servers do use a proxy cache for static HTML pages.

Palm.Net currently uses a proxy cache; the proxy servers pass each web clipping request on to the proxy cache after the server translates the request into an HTML(S)/HTML message request. Note that dynamically generated pages are not cached; this includes pages that end in or include file extensions such as `.jsp`, `.cfm`, `.cgi`, and `.shtml`.

Security Issues

The Viewer uses advanced security software to ensure that transmissions to and from Palm Powered handhelds are suitable for security-conscious applications such as electronic commerce.

Palm, Inc. uses strong cryptography, including the changing of session keys for every transmission to and from Palm Powered handhelds. Numerous security checks are employed in the logon process. Palm uses the elliptic curve cryptosystem, pioneered by Certicom, to encrypt wireless transmissions. This system features unparalleled security, efficient operation, quick speed, and small message size, and uses a 163-bit key that is equivalent in strength to RSA 1024-bit keys.

The Palm proxy servers use the high-strength, North American exportable version of SSL for data encryption and server authentication.

Fully redundant firewalls, with one-minute failover, provide network security for all Internet-accessible servers in the Palm.Net server farm. All Palm web clipping and IMessenger server functions run on secure computers.

Developer Support

There are several sources of on-line support for web clipping application developers:

- You can find Knowledge Base articles, white papers, sample code, and other documentation http://www.palmos.com/dev/tech/webclipping/.
- You can subscribe to the web clipping forums by following links from the Palm Developer Forums web page at http://www.palmos.com/dev/tech/support/forums/. There are two forums related to web clipping development:
  - The web clipping announcements forum, which features announcements from Palm, Inc. about web clippings, the Palm proxy servers, web clipping application software updates, and related items. Palm, Inc. strongly recommends that all current and future developers join this forum.

  - The web clipping forum, which allows you to post questions and share information with other web clipping application developers.

- You can download functioning web clippings for comparison at http://www.palm.net/apps/index.html.

Tools

To develop web clipping applications and web clipping, you need a few tools, as shown in Table 1.1.

Table 1.1 Tools for developing web clipping applications

Tool	Description
Text editor	Use to create your HTML pages. Although you can use an HTML editing tool to create your pages, Palm recommends against this, because most HTML editors add a lot of bulky code to their pages. If you do use an HTML editor, you need to clean up the generated HTML code to ensure that it conforms to the HTML 3.2 subset supported by the Viewer. Appendix B, "Palm HTML Summary," provides a summary description of the HTML that you can use in web clippings.
Graphics editor	Use to create the images that you include in your web clipping application. You can include GIF, and JPEG format images on your pages. You can use BMP format images for the icons used for your application in the Palm OS Applications Launcher.

Table 1.1 Tools for developing web clipping applications *(continued)*

Tool	Description
WCA Builder	The Palm web clipping applications builder program, which transforms your HTML and image files into a web clipping application (`.pqa`) database. You can download the Web Clipping Application Builder from the Palm Developer Web Clipping page, at http://www.palmos.com/dev/tech/webclipping/. This tool is free.
Palm Desktop Software	The software that is included when you purchase a Palm Powered handheld. Use this to install your web clipping applications on your handheld device.
POSE	The Palm OS Emulator program, which you can use to test and debug web clippings (and other Palm OS applications) on your desktop computer. This tool is free.
ROM images	A ROM image to use with POSE for testing your application. For more information, see Chapter 2, "Getting Started with Web Clippings."
Palm Powered handheld	You need a device that supports web clipping for testing and enjoying your web clipping applications. Palm VII devices support web clipping, as do other Palm Powered handhelds in conjunction with additional equipment, such as the Mobile Internet Kit and a wireless modem.

2

Getting Started with Web Clippings

This chapter provides a step-by-step guide to creating your first web clipping application. After following the steps in this chapter, you will have a functional web clipping application and you will know about:

- Creating a Simple Web Clipping HTML File
- Building and Running Your Web Clipping Application
- Testing Your Application With POSE
- Installing Your Application on a Palm Powered Handheld
- Adding Additional Content to Your Application

The final section in this chapter, Quick Start Summary, provides a one page summary of the steps you need to take to create a web clipping application.

After following the steps in this application, you can read the other chapters in this book to learn how to enhance and fine-tune your application. Specifically, Chapter 4, "Creating Web Clipping Applications," provides a wealth of information about adding various features to your web clipping applications.

Creating a Simple Web Clipping HTML File

The first step in creating a web clipping application is to create an HTML file on your desktop computer. Palm, Inc. recommends using a simple text editor such as NotePad or SimpleText for this purpose.

For our first example, you create a simple web page that displays the phrase "Hello World!" on the screen. Create a text file named `index.html` that contains the text shown in Listing 2.1.

NOTE: You do not have to name your top-level web page `index.html`; however, all of the examples in this chapter do so.

Listing 2.1 The Hello World web page

```
<html>
<head>
<title>My Web Clipping</title>
<meta name="PalmComputingPlatform" content="true">
<meta name="PalmLauncherRevision" content="1.0">
</head>
<body>
<h1 align="center">Hello World!</h1>
</body>
</html>
```

We make the text line a heading (`h1` tag) so that it displays in a large, bold font. For more information about the fonts you can use with different HTML tags, see Specific Attribute Values for Web Clipping in Chapter 6, "HTML for Web Clipping Applications."

Building and Running Your Web Clipping Application

Run the Palm Web Clipping Application Builder (WCA Builder) program to build your web clipping application. The examples in this chapter use the Windows version of the WCA Builder; the Macintosh version operates in a very similar manner. For more information about using the WCA Builder program, see Chapter 7, "Using the Web Clipping Application Builder."

NOTE: Web clipping applications were originally called Palm query applications; thus, WCA Builder creates web clipping applications with the `.pqa` extension.

Follow these steps to build your first web clipping application:

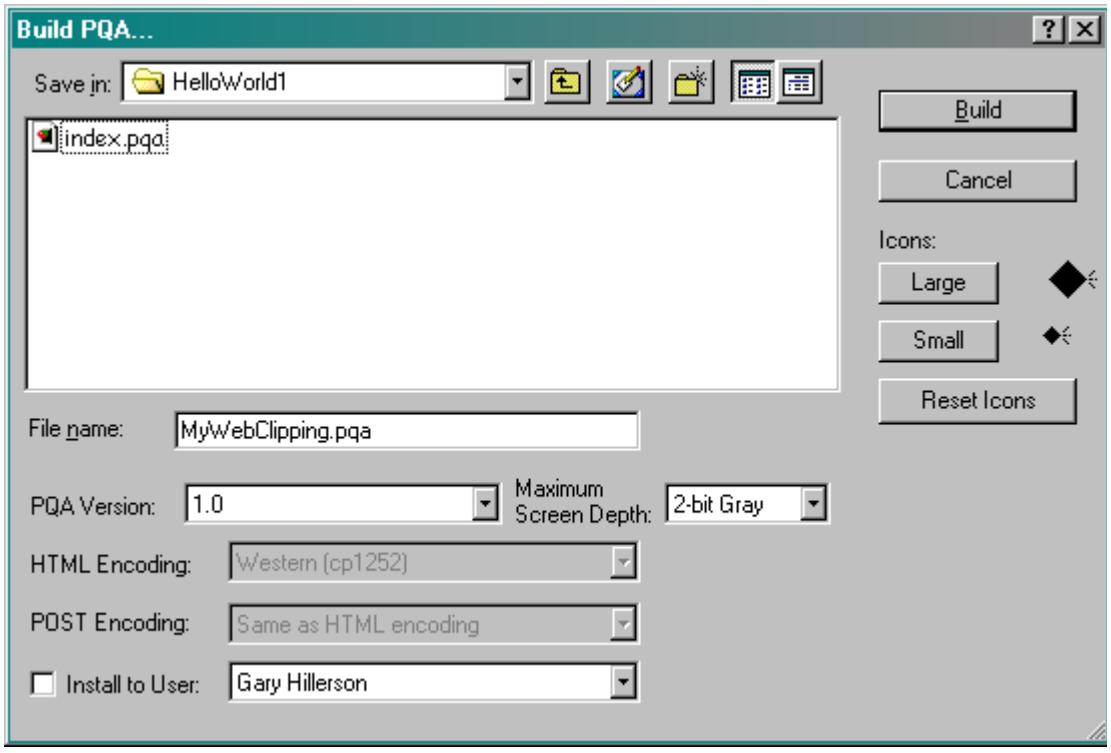
1. Start the `wcabuild.exe` program.

2. Select **Open Index** from the File menu.
3. Navigate to and open the `index.html` file that you created for this test. WCA Builder opens your HTML file.
4. Select the **Build PQA** command from the WCA Builder File menu.

   When the Build PQA dialog box displays, select the directory in which you want your `.pqa` file saved, and enter a name for the file. As shown in Figure 2.1, we have used "MyWebClipping" for our first example. For now, you don't need to enter or change any other values in the dialog box.

   After entering a name for your application file, click the **Build** button. The WCA Builder program builds your web clipping application and saves the `.pqa` file.

Figure 2.1 Building your web clipping application



You can test your application with the Palm OS Emulator (POSE) program, as described in Testing Your Application With POSE.

You can also install your application on a Palm Powered™ handheld and run it, as described in Installing Your Application on a Palm Powered Handheld.

Testing Your Application With POSE

The easiest way to test and debug your web clipping applications is to use them with the Palm OS Emulator (POSE) program, which runs on your personal computer and emulates different models of Palm Powered handhelds. When you run your web clipping application with POSE, it performs exactly as it will on a "real" Palm Powered handheld, including the manner in which it communicates with the Palm proxy servers and web sites on the Internet. Additionally, you can emulate different Palm devices and configurations with POSE, making it an almost indispensable tool for testing your web clipping applications.

NOTE: When you configure POSE to access Internet content, as described in this section, you can test net access at high speed and without the additional cost of using the Palm.net service.

This section shows you how to run your application with POSE. For more information about POSE, see the *Palm OS® Programming Development Tools Guide*.

You need to follow these steps to run your application with POSE:

1. Download POSE to your personal computer.
2. Install POSE and the ROM Transfer tool that is packaged with POSE.
3. Start POSE and transfer the ROM from a Palm Powered handheld that supports web clippings, such as the Palm VII™.

   Alternatively, you can download ROM images from the Palm Solution Provider web site, as described below.
4. Configure POSE to use the correct proxy server setting for web clipping to work.
5. Install your web clipping application with POSE.

6. Test and debug your web clipping application.

The following sections describe these steps in detail.

1. Download POSE

The most recent released version of Palm OS Emulator (POSE) for the Macintosh, Windows and Unix is always posted on the Internet in the Palm developer tools area:

http://www.palmos.com/dev/tech/tools/

Follow the links to the Palm OS Emulator page to retrieve the released version of the Emulator. The package that you download also includes the ROM Transfer program that you use in the next step.

2. Install POSE and the ROM Transfer Tool

The POSE package that you download contains the POSE executable application file, which you move to your directory of choice on your personal computer. To run POSE, simply double-click the executable file, which is named as shown in the following table.

Operating system	POSE executable file name
Windows	`Emulator.exe`
Macintosh	`Palm OS Emulator`
Unix	`pose`

NOTE: When you download POSE for Unix, you get the source code, which you must compile on your own.

When you download POSE, you also receive the ROM Transfer program, which allows you to transfer a memory image of the Read Only Memory (ROM) software in a Palm Powered handheld. POSE uses this ROM image to emulate the handheld device. If you have

not previously installed the ROM Transfer tool, follow these steps to do so:

1. Use the Install tool that is part of the Palm Desktop organizer software to set up the installation of the `ROM Transfer.prc` file that was packaged with POSE.
2. Use the HotSync® Manager to synchronize with your handheld to install the ROM Transfer program.

3. Transfer a ROM Image

You can use one of the following methods to transfer a ROM image for POSE to emulate:

- Use the ROM Transfer tool to transfer a ROM image from a hardware device that you have in your possession, as described in Transfer a ROM Image Using the ROM Transfer Tool.
- Download a ROM image from the Palm Resource Pavilion and start a new POSE session that uses that image, as described in Download a ROM Image From the Solution Provider Program.

NOTE: Some users experience difficulty using the ROM Transfer program with their handheld devices. If you have this problem, you need to download a ROM image from the Palm Resource Pavilion.

Once you have transferred a ROM image, the POSE display looks just like the Palm Powered handheld that it is emulating. Figure 2.2 shows the POSE display for the Application Launcher in an emulated Palm VII.

Figure 2.2 The POSE display

Transfer a ROM Image Using the ROM Transfer Tool

To use the ROM Transfer tool, first install POSE and the ROM Transfer tool on your personal computer, and then run POSE.

If you are running POSE for the first time, you need to transfer a ROM image for POSE to use. Follow the instructions shown in Table 2.1 and make sure that you transfer a ROM image from a device that supports web clipping, such as the Palm VII.

Table 2.1 Running POSE for the first time

Operating system	Instructions
Windows	Choose the **Download a ROM image from a Palm OS device** command from the initial dialog box, and follow directions.
Macintosh	Dismiss the initial dialog box, choose the **Transfer ROM** command from the File menu, and follow directions.
Unix	Right-click the POSE display, select the **Transfer ROM** command, and follow directions.

If you are not running POSE for the first time, transfer a ROM image as follows:

- in Windows or Unix, right-click the POSE display and choosing the **Transfer ROM** command
- on a Macintosh, choose the **Transfer ROM** command from the File menu

Download a ROM Image From the Solution Provider Program

You can download ROM images for any of the devices sold by Palm, Inc. from the Palm Resource Pavilion. To use a ROM image in POSE, follow these steps:

1. After you have downloaded the ROM image, start POSE.
2. Choose New from the POSE menu, to start a new session.
3. Select the ROM image that you just downloaded.

POSE starts a new session, emulating the ROM that you just downloaded.

NOTE: When you subsequently start POSE, it automatically reloads your previous session and ROM image. If your Caps Lock key is down (enabled) when POSE starts, it instead presents a dialog box that allows you to choose a different ROM image or start a new session. This is convenient when your most recent session had problems.

To access the Resource Pavilion, you must join the Palm Solution Provider program. Follow these steps:

1. Go to the Solution Provider Program web page, at:

   http://www.palmos.com/dev/program/

2. Click the **Join Now** link.
3. Fill out the application and submit it.

   It takes approximately two days for Palm to approve your application, at which time you will receive an email that describes how to log into the Resource Pavilion.

4. Click the ROM Image Clickwrap Area.
5. Read and accept the license agreement.
6. Download the ROM image(s) that you want. If you are downloading a ROM image to test your web clippings, make sure that you select a ROM that supports web clipping.

NOTE: If you live in certain parts of the world, access to some areas of the Resource Pavilion and to some of the ROM images is not allowed, due to United States government regulations. This is because the Palm VII ROMs include encryption that cannot be exported to certain countries.

Due to these regulations, when you click the link to download certain ROM images, the Resource Pavilion verifies your location. You may notice a delay before the download begins. And if your location cannot be verified using a reverse DNS lookup, then you may have to fill out and mail in additional forms to receive the Palm VII ROM images. If you are a member of the Palm Development Seeding Program, you do not have to fill out these forms to access the ROM images.

4. Configure POSE for Web Clippings

After you have transferred a ROM image to your personal computer for use with POSE, you need to enable POSE for web clipping, which requires you to change two settings in POSE:

- Redirect NetLib requests to your desktop computer's TCP/IP connection, as described in Redirecting NetLib Requests in POSE.
- Change the proxy server setting in POSE, as described in Change the Proxy Server in POSE.

Redirecting NetLib Requests in POSE

Follow these steps to redirect NetLib requests in POSE:

1. If you're using Windows or Unix, right-click the POSE display and select **Preferences** from the Settings menu. If you're on a Macintosh, select **Preferences** from the Edit menu.
2. Check the **Redirect NetLib calls to host TCP/IP** checkbox.

3. Reset the emulated device: in Windows or Unix, right-click and choose **Reset**; on a Macintosh, select **Reset** from the File menu.

Change the Proxy Server in POSE

You need to change the proxy server setting in the emulated device to make web clippings work with POSE. Follow these steps in POSE:

1. Click the Applications Launcher button.
2. Click the Prefs application.
3. Select **Web Clipping** from the pulldown menu.

NOTE: In versions of the Palm OS® earlier than version 4.0, you need to select the Wireless preferences instead of the Web Clipping preferences.

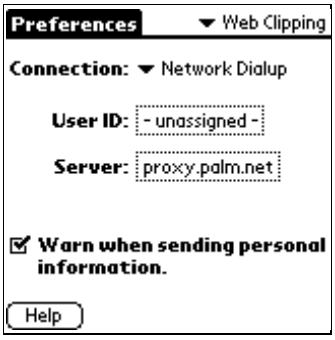
---

4. Set the Proxy server value.

   If you are using version 4.0 or later of the Palm OS, you can use the `proxy.palm.net` alias for the proxy server value, as shown in Figure 2.3.

   If you are using an earlier version of the Palm OS, enter an IP address value. You can determine the current proxy server IP address to use by checking http://oasis.palm.com/dev/proxy/.

Figure 2.3 Setting the proxy server value in POSE



IMPORTANT: The IP address to use for the Palm proxy servers changes from time to time. The safest means of maintaining a correct address is to use the `proxy.palm.net` alias with Palm OS version 4 or later, as shown in Figure 2.3.

If you are using a version of the Palm OS earlier than version 4.0, look up the latest proxy server addresses at http://oasis.palm.com/dev/proxy.

5. Install Your Web Clipping Application in POSE

To install your web clipping application for testing with POSE, select the **Install Application/Database** command and then navigate to the location on your personal computer where you saved your web clipping application (`.pqa`) file. Select your file, and then click the **Open** button in the dialog box. POSE installs your application in the current emulation session.

NOTE: When the Web Clipping Application Builder builds your application, it automatically sets the category for your application to the `Palm.Net` category.

If you are using Windows or Unix, right-click the POSE display, and select the **Install Application/Database** command. If you are on a Macintosh, select the **Install Application/Database** command from the File menu.

Note that you can also drag and drop your `.pqa` file into the POSE window to install it.

If you install your application while the Application Launcher is active in the emulated device, POSE does not automatically refresh the display to show the new application in the Launcher. To refresh the display, switch to an application or perform a Reset, and then return to the Application Launcher. Your web clipping application is now listed in the Application Launcher.

Another way to ensure that your application shows up immediately is to switch to a different application category in the launcher before installing your application, and then switch to the Palm.net category.

6. Run Your Application

You run your application in POSE just as you do on a Palm Powered handheld hardware device:

1. Tap the **Applications** button to display the Applications launcher.
2. Tap the name or icon for your web clipping application. The Web Clipping Application Viewer (the Viewer) runs your web clipping application.

When you run your first web clipping application, the POSE screen displays "Hello World," as shown in Figure 2.4.

Figure 2.4 Running your first web clipping application

Note that the title bar of the web clipping application shows the title that you code in your HTML page. The Viewer automatically truncates the title to fit on the Palm screen.

NOTE: The Viewer allocates 74 pixels for displaying the application title. If the display of your title is longer than 74 pixels, the Viewer displays an ellipsis (...) at the end of the title bar.

The user can display the entire title by choosing the Info item from the Options menu. In versions of the Palm OS earlier than version 3.5, users can display the entire title by tapping on the title bar.

Installing Your Application on a Palm Powered Handheld

Once you have tested and debugged your web clipping application with POSE, you might want to install and try it on your own web clipping capable handheld device. You can install your application in exactly the same manner as you install any other application. Follow these steps:

1. Run the Install Tool that came with the Palm Desktop Software for your device.
2. Select your web clipping application (`.pqa`) file for installation.
3. Synchronize your handheld device.

You can now run your web clipping application on your handheld device.

Adding Additional Content to Your Application

Now that you know how to create and test a web clipping application, you can add more content to the application. This section provides instructions for adding a few, simple enhancements to your web clipping application:

- Displaying a Clipping From the Internet
- Adding a Button to Your Application
- Adding Images to Your Application

You can add many other capabilities to your web clipping applications, including:

- using advanced forms
- sending email
- launching another application on the user's handheld device
- downloading files from the Internet
- using information about the user's location in queries

For more information these and other possibilities, see Chapter 4, "Creating Web Clipping Applications."

IMPORTANT: The content area for web clippings is 153x144 pixels. This limits both the amount of text you can display without scrolling and the size of images.

Displaying a Clipping From the Internet

The first obvious addition for the sample web clipping application is to actually retrieve a web clipping from the Internet. This requires two operations:

- adding a link to the Hello World application
- creating web clipping content at the linked-to address on the Internet

The computer on which you are running POSE to test your web clipping or the Palm Powered handheld with which you are testing must be connected to the Internet to retrieve and display the web clipping.

Adding a Link to the Example Program

You can use standard HTML code to add a link to the sample application, as shown in the paragraph (...) added to our example code in Listing 2.2.

Listing 2.2 The Hello World web page with a link

```
<html>
<head>
<title>My Web Clipping</title>
<meta name="PalmComputingPlatform" content="true">
<meta name="PalmLauncherRevision" content="1.0">
</head>
<body>
<h1 align="center">Hello World!</h1>

<p>Tap this <a href="http://oasis.palm.com/devsupp/html/
hw_clip.html">link</a> to retrieve a clipping.</p>
</body>
</html>
```

The display of the new application is shown in Listing 2.2. Note that the Viewer automatically underlines and adds the over-the air ( ) image to the link anchor.

Figure 2.5 The Hello World application with a link

Creating a Web Clipping

The other job that you have to complete to make retrieving a web clipping possible is to create that clipping. In most cases, web clipping applications retrieve a web clipping that has been dynamically created by a CGI script or an application on the server; however, for our simple example, we retrieve a static page that is stored on a server at Palm. Listing 2.3 shows the contents of the web clipping that is stored in the `hw_clip.html` file on the Palm server.

Listing 2.3 The Hello World web clipping

```
<html>
<head>
<meta name="palmcomputingplatform" content="true">
<meta name="historylisttext" content="HW Clip &date &time">
<title>HW Clipping</title>
</head>
<body>
<h1 align="center">Hello Back From Palm!</h1>
<br><br><br><br><br><br><br><br>
<p><a href="file:MyWebClipping.pqa" button>Home</a></p>

</body>
</html>
```

Note that the Home button in this clipping links to our example's home page, which is the `MyWebClipping.pqa` file. Also note that the Home link uses the Palm `button` attribute, which is described in the next section. When this clipping is retrieved, it displays as shown in Figure 2.6.

Figure 2.6 The retrieved web clipping

IMPORTANT: A primary consideration in any web clipping page that gets sent from the Internet is the amount of data in the HTML page: the larger the page, the more data that has to be sent over a relatively slow communications link. Whenever possible, make choices in your web clipping pages that minimize the page size. This includes reducing the number of attribute changes on a page, each of which requires the transmittal of data.

Adding a Button to Your Application

Web clippings can use several HTML attributes that Palm, Inc. has added to the Viewer. These attributes are fully described in Chapter 6, "HTML for Web Clipping Applications."

You can use the `button` attribute to make a link into a rounded rectangular button. The text for the link is displayed inside of the rounded rectangle.

For example, you can add the button attribute to the link in our sample application. Simply change the link paragraph to the following:

```
<p>Tap this <a href="http://oasis.palm.com
devsupp/html/hw_clip.html" button>link</a> to
retrieve a clipping.</p>
```

Now our web clipping application displays as shown in Figure 2.7

Figure 2.7 Changing the link to a button

Adding Images to Your Application

You can add images to your web clipping applications and to the web clipping pages that you retrieve from the Internet.

You can use the HTML `<img>` tag to add an image, as shown in the line added to our example code in Listing 2.4. This example displays the Palm logo, centered below the heading line, with some space above and below the logo.

Listing 2.4 Adding an image to the Hello World application

```
<html>
<head>
<title>My Web Clipping</title>
<meta name="PalmComputingPlatform" content="true">
<meta name="PalmLauncherRevision" content="1.0">
</head>
<body>
<h1 align="center">Hello World!</h1>

<br><br><br>
<center>
<img src="palmlogo.gif">
</center>
<br><br><br>
```

```
<p>Tap this <a href="http://oasis.palm.com/devsupp/html/
hw_clip.html" button>link</a> to retrieve a clipping.</p>

</body>
</html>
```

You can substitute a graphic image file of your choice for the `palmlogo.gif` file that we have used. The resulting web clipping application displays as shown in Figure 2.8.

Figure 2.8 The Hello World application with an image

Remember that the content area is 153 pixels wide by 144 pixels high when you design images for web clippings.

For more examples of using images on web clipping pages, see Using Images in Web Clippings in Chapter 4, "Creating Web Clipping Applications."

For more information about using the `<img>` tag in your web clipping pages, see Images on Web Clipping Pages in Chapter 6, "HTML for Web Clipping Applications."

For more information about customizing images for use in your web clipping pages, see Using the Image Checker Program in Chapter 7, "Using the Web Clipping Application Builder."

Quick Start Summary

This section provides a summary of the operations you need to perform to create and test a web clipping application:

1. Create your web clipping HTML and image files. Name your top level page `index.html`. If you use a WYSIWYG HTML editing program, make sure that you strip out any extra tags added by the program.
2. Build your web clipping application with the Palm Web Clipping Application Builder (WCA Builder) program:
   a. Start the WCA Builder program. On Windows, the executable is named `wcabuild.exe`.
   b. Choose the **Open Index** command from the File menu, and select your `index.html` file.
   c. Choose the **Build PQA** command from the File menu, and select the destination for your built web clipping application (`.pqa`) file.
3. Test your web clipping application with POSE:
   a. Start the POSE program.
   b. Make sure that you have loaded a ROM image for a Palm Powered handheld that supports web clipping, such as the Palm VII.
   c. Check the **Redirect NetLib call to host TCP/IP** box in the POSE preferences.
   d. Set the proxy server address in the emulated device's Wireless preferences. You can determine the current proxy server IP address to use by checking http://oasis.palm.com/dev/proxy/.
4. Choose the **Install Application/Database** command in the POSE menu, and select your `.pqa` file for installation into the emulated device session.
5. Tap on the Application Launcher icon in POSE, and then tap on the icon/name that you chose for your web clipping application. Your web clipping application launches.
6. After you have debugged your web clipping application, you can install it on your handheld with the Palm Desktop Software Install tool, just as you would install any other application.

3

Web Clipping Style Guidelines

This chapter provides design and style guidelines for web clipping applications. Although you develop web clipping applications with standard HTML, you need to ensure that the web pages you create for display on Palm Powered™ handhelds provide the user with a quality experience.

Part of the specialized design of web clipping applications is due to the small screen size with which users are interfacing with the Internet. Another difference between web clipping applications and desktop browsing is that web clippings are designed to provide information on demand to mobile users. This is, by design, more focused than the usual desktop browsing experience.

This chapter presents the information you need to make design decisions for your web clipping applications in the following sections:

- Overview of Web Clipping Design presents an overview of the user experience of web clippings and the design constraints associated with that experience.
- Design Considerations describes specific considerations for designing your web clipping applications.
- Web Clipping Application Structure provides a high-level view of the structure of web clipping applications.

Overview of Web Clipping Design

The mechanics of authoring application pages and clippings differ very little from the mechanics of authoring web pages and CGI scripts or Java servlets in general. However, you need to adapt your design philosophy to take into consideration the constraints of a

small, handheld screen that is connected to the Internet with a relatively expensive and slow link.

Specifically, you do need to pay a lot of attention to economizing on connect time and web file size. Because of this, the web clipping model is designed to provide access to targeted bits of Internet information.

The typical web clipping user does not focus on following hyperlinks to access information. Instead, the user composes and sends a very specific query from the web clipping application to a site that responds to that query. For example, the user might send a request for a stock quote, or for the current weather in a specific city.

Remember that the user is paying for transmission on a per-byte basis, which means that every byte sent to or from your web clipping application counts. The basic Palm.Net plan allows transmission of 50 KB per month, which is quickly exhausted if your applications are not economizing on transmission size.

One thing you can do to reduce transmission requirements is to store HTML pages and graphics locally on the handheld device. Another is to reduce the number of attribute changes and extraneous HTML coding on your pages.

The remainder of this chapter provides specific design considerations and techniques for creating web clipping applications that provide users with economical and efficient web browsing on their Palm Powered handhelds.

Design Considerations

This section provides you with specific information about the following design issues and considerations:

- Design for Your Users
- Design for a Small Screen
- Design for Speedy Display
- Decide Which Information Goes Where

Design for Your Users

The primary goal of your design must be to target the people who will be using your software. Specifically, you need to keep focused on the fact that the mobile browsing experience is inherently different than the desktop browsing experience, and that your users will want to quickly and easily get to the information they are seeking. To put is succinctly, you want to keep your web clipping applications lean, quick, and highly focused.

Palm, Inc. suggests that you concentrate on the following design issues:

- Determine your target audience. If possible, survey the audience to determine their most important needs and prioritize those needs.
- Determine the vital content for your application and stick with effectively presenting that content. Remember that web clipping applications are intended to present clippings, like showing someone a specific article from a newspaper.
- Implement only absolutely required features, and make sure that the most frequently used features are the most easily accessible ones.
- Design your page(s) to help users formulate precise queries that generate small and highly focused results pages.
- Present only the most essential information to the user.
- Let "real" users test your web clipping application and listen to their feedback.

Design for a Small Screen

The screen on Palm Powered handhelds and other mobile devices are quite a bit smaller than even the lowest resolution desktop computer screens. To provide the user with a reasonable browsing experience on the small screen, you must accommodate this limitation. Palm, Inc. suggests the following design tips:

- Limit the need for scrolling. Although version 2.0 of the Web Clipping Application Viewer (the Viewer) allows scrolling, Palm strongly suggests that you not require users to scroll their screens. Scrolling can be awkward, especially in some of the situations in which users are accessing content on their

handheld devices. Although you may not be able to eliminate scrolling of web clipping pages returned from a server, you can (and should) eliminate scrolling in your static pages.

- Limit the width of tables in your pages. Again, requiring a user to scroll across the columns or down the rows of a table to view all of the data makes the viewing experience less effective.
- Limit the size of graphic images to fit onto the screen.
- Organize your application so that your index page has links to non-vital information, rather than trying to fit it onto the page. For example, provide a link to your About page, instead of including that information on your index page.

Design for Speedy Display

The most important aspect of a quality browsing experience for users is speed. You must design your application to return results to users as quickly as possible. Here are some specific guidelines to help improve the speed of your web clipping applications:

- Structure your web clipping results pages to return the most focused content possible. Design the pages to minimize the frequency with which the user needs to tap links to access online content.
- Whenever possible, link to local—on-device—content to present additional information. This content is bundled into your `.pqa` file, and thus displays very quickly.
- Store graphics for clippings locally in the application, rather than transmitting them over the air.
- You can use the `localicon` meta tag in your web clipping application pages to force the Web Clipping Application Builder tool to include in your `.pqa` file content that is not referenced in your pages.
- If you are designing the web clipping results pages, you can have those pages refer to images and content that are stored in your `.pqa` by using the `file:` designator in the HTML page.
- Resort to over-the-air hyperlinks only when absolutely necessary.

Studies reveal that users consistently choose faster applications over slower ones, even if the slower application has additional features. And ergonomic experts have noted that users notice even a one-half second delay. So any efforts you can make to improve the speed of your application will increase its acceptance among users.

For more information on including local content and using hyperlinks in your web clipping applications, see Chapter 4, "Creating Web Clipping Applications."

Decide Which Information Goes Where

When you create a standard web page, all of the information is stored on a server that is connected to the Internet. When you design a web clipping application, some of the information is stored on a server and some of the information is stored in the web clipping application itself.

This means that you need to decide which information is stored locally on the Palm Powered handheld and which is stored remotely, on an Internet server. Palm, Inc. recommends that you follow these guidelines:

- Include the more stable content in your web clipping application, including the query form, image files, and frequently consulted, nonvolatile information.
- Store the more volatile information on the Internet server. This information, which changes frequently, is downloaded as a web clipping upon request, and then is displayed by your application.
- Remember that whatever you store in your web clipping application remains static until the user upgrades your application.
- Remember that whatever is accessed online takes a relatively long time to arrive at and be displayed on the user's device.

Web Clipping Application Structure

Your web clipping applications must have a top-level page, which is usually given the file name `index.html`. Some applications consist

of only this top-level page, while others are organized into multiple pages with the top-level page at the root.

The remainder of this section provides descriptions and shows examples of both single-page and multiple-page web clipping applications.

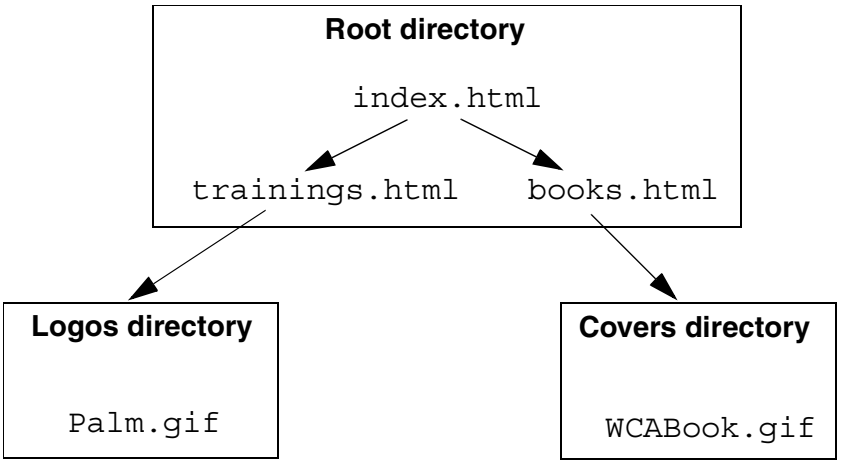
Storing Your Files

You create HTML and graphics files from which to build your web clipping application. The Web Clipping Application builder requires that all of these files reside in the same directory hierarchy.

As with standard web design, you create a home page for your web clipping. Most developers use the name `index.html` for their home pages, though you can use whatever name your choose. Your home page provides the query form with which the user interfaces, or provides links to the principal content of your web.

Figure 3.1 shows the web directory structure for a simple web that includes two image subdirectories.

Figure 3.1 Sample web directory structure



Hyperlinking to Local Content

Hyperlinks to local content—other pages and images in your web—are treated differently in your web clipping applications than they are in the web clippings returned to your application, as follows:

- Within the web on your desktop computer from which you build your web clipping application, links to other pages and images must include a relative path name, just as they would in any web.

  For example, using the directory structure in Figure 3.1, you would use the following code to refer to the `Palm.gif` image:

  ```
  <img src=Logos/Palm.gif>
  ```

- In the web clippings sent back to your application, all links to local content must be made without including a path name. Thus, if a web clipping wants to refer to the same image, it must use the following code:

  ```
  <img src=file:myapp.pqa/Palm.gif>
  ```

The reason for this difference is that when the Web Clipping Application Builder (*WCA Builder*) creates your application, it converts each page and image into records in the database stored in the generated `.pqa` file. During the build process, the WCA Builder tool resolves all of the path name references.

WARNING! Since all of the files in your web are stored as records in a single database within the `.pqa` file, every file name must be unique. You cannot rely on the file name suffix (such as `.html` or `.gif`) to make the name unique.

For more information about images in your web clippings and applications, see Using Images in Web Clippings.

Single-Page Web Clipping Applications

Many web clipping applications consist of a single page that provides a query form. The user composes a query and submits it for over-the-air transmission. A results web clipping is returned from the Internet, and the application displays the clipping on the

handheld screen. The structure of such an application is very simple, consisting of a single `index.html` page.

Other single-page web clipping applications consist of a set of hyperlinks, each of which causes the download of a web clipping from a location on the Internet.

For example, Figure 3.2 shows a simple web clipping application that allows the user to look up addresses. Note that, although this application is considered a one page application, it does include additional pages that display help and legal information.

Figure 3.2 A Simple web clipping application

Multipage Web Clipping Applications

Your web clipping application might also be structured into multiple pages and images that are stored on the Palm Powered handheld. And the results pages in your application might link to:

- local pages on the handheld
- local images
- other Palm OS® applications on the handheld
- URLs on the Internet

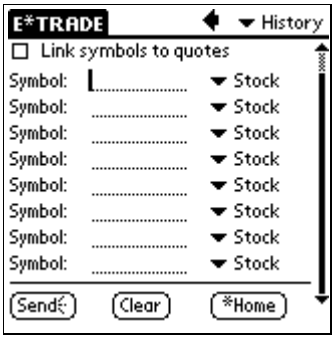
Figure 3.3 shows an example of a multipage web clipping application.

Figure 3.3 Multipage web clipping application

Palm, Inc. recommends that you keep the hierarchy of your web clipping applications as shallow as possible. Keep the following points in mind when designing your application:

- Make it easy for users to access the information without drilling down through too many layers.
- Remember that users of wireless, handheld devices tend to be much less patient in their web browsing activity than do desktop browser users.
- Also make it easy to move back up the information hierarchy: to a previously visited page, or to the top-level index page.
- Include a Home button in your application to allow the user to quickly return to your top-level page. Figure 3.4 shows a web clipping application screen that includes the Home button.

Figure 3.4 Application with home button



- Encourage the user to use the History list, which is a cache of recently used pages for each application that the user can access at the top of the application display area on the handheld screen. Figure 3.4 shows the History List access arrow; when the user taps on this arrow, the history list for that application is displayed, as shown in Figure 3.5.

Figure 3.5 Displaying the history list for an application

Each web clipping can include the `HistoryListText` META tag, which provides a custom string to identify that clipping in the history list. For more information, see The HistoryListText Meta Tag in Chapter 6, "HTML for Web Clipping Applications."

Legal Implications for Web Clipping Application Developers

Palm, Inc. provides the following guidelines on copyright issues associated with the use of web clipping technology. These guidelines **do not** constitute legal advice. Palm, Inc. strongly encourages you to consult a copyright attorney for answers to any legal questions you have.

- Web clipping technology, which entails retrieving information from the World Wide Web, could leave the door open to inadvertent copyright infringement. As a developer of web clipping applications, you must make every effort to prevent such infringement.
- Retrieving content available on the web and redistributing that content without the express permission given by the content owner is likely to constitute copyright infringement. This may be true whether or not you provide attribution and whether or not information or software is distributed for free.

- Before clipping web content and providing it to Palm device users, you must first get the permission of the copyright owner. Note that the owner of the web site from which material is to be retrieved may not be the owner of the content you wish to use. That web site itself may only have a license to use the content and not the right to sublicense that content to other sites.

Make sure you have all necessary rights to reproduce, modify, display or distribute any material subject to any third party copyrights or other intellectual property rights.

4

Creating Web Clipping Applications

This chapter describes specific topics related to creating web clipping applications. It expands upon the information provided in Chapter 2, "Getting Started with Web Clippings."

This chapter describes the following topics:

- Creating Multi-Platform HTML
- Linking to Local Files
- Using Images in Web Clippings
- Using Forms in Web Clipping Applications
- Publishing Your Web Clipping Application

For information about more advanced topics, see Chapter 5, "Advanced Web Clipping Actions," which describes how to send mail, download and upload files, and launch Palm OS® applications from web clipping applications.

Creating Multi-Platform HTML

You can use the `smallscreenignore` HTML tag to create HTML pages that display differently on different platforms. This tag allows you to specify that portions of the page are ignored when the HTML page is viewed on a Palm Powered™ handheld.

When returning HTML pages back to the device, the Palm proxy servers strip out any content that is found between the `<smallscreenignore>` and `</smallscreenignore>` tags. For example, if your online page includes the following code:

```
<smallscreenignore>
<img src="bigpic.gif" height="600" width="450">
</smallscreenignore>
```

Then the `bigpic.gif` image displays when the page is viewed on a desktop browser, but not when the page is viewed in your web clipping application.

Note that the proxy server strips out the ignored content; thus, there are no speed consequences for web clipping applications receiving the page.

Linking to Local Files

One simple way to reduce the amount of data that needs to be transmitted from the Palm proxy servers to the user's Palm Powered handheld is to use local content, which is data stored in your web clipping application (`.pqa`) file. You can refer to this content—images or HTML file—in your links.

Including Content in Your Application File

If any page in your local web refers to local content, the Web Clipping Application Builder (WCA Builder) program automatically builds includes the referred-to page or image in your application file. WCA Builder does this by traversing your entire web, following the links it finds:

- if a link on your index page refers to a local page, that file is bound into your built application, and its links are followed
- if a link on any included page refers to a local file, that file is bound into your built application, and its links are followed
- if a link on any page refers to a local image, that image file is bound into your built application

You can also use the Palm `LOCALICON` meta tag to force WCA Builder to bind in images or pages that are not referred to by other pages. When WCA Builder finds the `LOCALICON` meta tag in a page, it treats the content portion of the tag just like as it does a local file (HTML or image) that is linked to by another page in your web.

The following is an example of using the `LOCALICON` meta tag:

```
<META NAME="LocalIcon" CONTENT="MyPhoto.GIF">
```

For more information, see The LocalIcon Meta Tag in Chapter 6, "HTML for Web Clipping Applications."

Referring to Local Files

To refer to a local image or HTML page in an URL, you use the `file` designator. Specify the name of your `.pqa` file as the directory name, and specify the image or HTML name as the file name. For example, to refer to the image file shown in the previous example, you would use the following HTML statement:

```
<IMG SRC="file:myapp.pqa/myphoto.gif">
```

You can use similar syntax to refer to a local HTML page:

```
<a href="file:mypqa.pqa/personal.html">My
Personal Page</a>
```

NOTE: You can use links to local content in both your local pages and in web clippings returned from the Internet. In fact, Palm, Inc. strongly encourages you to link to local content whenever possible, since this can dramatically reduce the amount of data that has to be transmitted over the wireless connection.

Using a Directory Hierarchy for Your Web Clipping Application

Web clipping applications do not support hierarchical file structures. However, you can organize your HTML and image files hierarchically on your personal computer. When you build your web clipping application, the WCA Builder program flattens the hierarchy and translates all links so that they work.

This has a few very important implications:

- You can use hierarchical references to files in your local web when working on your personal computer.
- You cannot use hierarchical references to files in web clippings that are sent back from the Internet. Instead, you

refer to all files as if they reside in the root directory of your application.

- All file names in the hierarchy must be unique, without relying on the hierarchy to create that uniqueness.

For example, suppose that, on your personal computer, your application's root directory includes your index page and two subdirectories, `images` and `content`, and that you are linking to file named `myphoto.gif` that is stored in the `images` directory.

In your index page, you can refer to that file as follows:

```
<IMG SRC="images/myphoto.gif">
```

When WCA Builder builds your web clipping application, it binds in your image file and modifies that link to remove the directory reference. Thus, if your application is named `myapp.pqa`, then WCA Builder changes the above reference to:

```
<IMG SRC="file:myapp.pqa/myphoto.gif">
```

IMPORTANT: Remember that, since WCA Builder is flattening the hierarchy, you cannot rely on your directory structure to make file names unique. If you have a file named `myphoto.gif` in both your `content` and `images` directories in the above example, the flattened reference would be ambiguous.

Using Images in Web Clippings

This section provides information about including images in your web clippings.

IMPORTANT: A primary consideration in any web clipping page that gets sent from the Internet is the size of the page: the larger the page, the more data that has to be sent over a relatively slow communications link. Graphic images, especially color images, can add a lot of overhead to your pages. Whenever possible, have your web clipping pages refer to images that are stored locally in your `.pqa` file.

You include images on your application and clipping pages just like you do in standard HTML pages, with the `IMG` tag. For example:

```
<IMG SRC="images/myphoto.gif">
```

Image Attributes in Web Clipping Pages

You can use most of the standard image attributes in images that you include on your web clipping pages:

- `ALT`
- `BORDER`
- `HSPACE`
- `VSPACE`
- `HEIGHT`
- `WIDTH`

The Viewer does not support image mapping; thus, you cannot use the following image attributes in your web clipping applications or web clippings:

- `ISMAP`
- `USEMAP`

NOTE: Remember that each attribute you specify increases the amount of HTML code that must be transmitted between the handheld device and the proxy server.

Image Borders

The Web Clipping Application Viewer (the Viewer) treats image borders differently than many browsers. You need to be aware of the following issues:

- The default border value is 0.
- To display a border, you must specify an even number value.
- Borders are always drawn as one pixel wide.
- The Viewer does not automatically draw a border around an image that is part of an anchor tag.

Image Restrictions in Web Clipping Pages

There are several important aspects of image display in web clipping applications that you need to know:

New Feature

- Starting with version 4.0 of the Palm OS, the Viewer allows horizontal scrolling of images. Image scrolling only works, however, if the page includes the `PalmComputingPlatform` meta tag.

  In earlier versions, the maximum width of an image on the Palm VII™ and similar Palm Powered handhelds was 153 pixels, and if you attempted to include an image that wider than 153 pixels, the Web Clipping Application Builder generated an error.

- The height of the content area on Palm Powered handhelds is 144 pixels. If you include an image that is taller than 144 pixels, the Viewer displays it and allows the user to scroll down to see the remainder of the image. However, Palm, Inc. strongly discourages this behavior, because requiring the user to scroll lessens the quality of his or her experience.

Managing Different Image Depths

New Feature

Starting with version 4.0 of Palm OS, you can include images with varying bit depths in your web clipping pages.

Of course, the type of Palm Powered handheld that the user has impacts how each image displays. In particular, if the user's device is running version 4.0 or later of the Palm OS, then the device reports its bit depth to the proxy server when sending a request. And the proxy server maps images into the appropriate bit depth for the device. For example, the proxy server will scale a 16-bit color image down to 2-bit or 4-bit grayscale if sending the image to an older Palm device.

NOTE: Palm Powered handhelds running version 4.0 or later of the Palm OS send `%WCDevCaps` information to the proxy server so that the server can determine how best to send information back. For more information, see The %WCDevCaps Variable in Chapter 6, "HTML for Web Clipping Applications."

The Viewer uses the Netscape, web-safe palette to display color images.

Managing Cached Pages and Hash Coded Links

When a Palm proxy server receives a web clipping from the Internet, it translates and compresses the clipping into CML before sending it to the Palm Powered handheld (the device). One of the operations that the server performs in this process is to replace URLs with index values, which allows the server to send data more efficiently to the device. The indexing process works as follows:

- Each link in the clipping is represented as a positional index. For example, the first link is assigned an index value of 1, the second link an index value of 2, and so forth.
- A checksum key is created from the full URL of each link. The full URL includes the protocol scheme and any parameter data included with the URL specification. The checksum is based on summing the ASCII character values of the URL.
- The proxy server sends the index value and key for each link in the clipping to the device, instead of sending the full URL for each link. The proxy server also sends exactly one instance of the web clipping page URL that contains the links to the device.
- Then, when the user taps a link in a clipping, the Viewer on the device sends the URL of the clipping that contained the link and the index and checksum for the link to the proxy server.

- The proxy server uses the URL to retrieve the original clipping, and uses the index to find the link in the page.

  The proxy server computes a checksum for the link and compares it to the original checksum. If the checksum comparison succeeds, then the proxy server follows the link, retrieves the page, and sends it back to the device. If the checksum fails, the proxy server generates an error message.

Unfortunately, this scheme can become problematic with dynamic clippings. If the clipping has changed since the links were indexed, then it is possible that the links have changed position in the clipping, which then results in the user seeing an error message. An additional complication is that the checksum computation method allows different URLs to generate the same checksum value, which means that the proxy server can miss the fact that an URL has changed.

Some content providers work around this problem by having their clippings contain the same links that point to pages that are updated periodically on the server. Other solutions involve having each link in a clipping include a session ID value such as the `%deviceId` value.

New Feature

Starting with version 4.0 of the Palm OS, you can use the `PalmHREFStyle` meta tag to specify that you want the Viewer to receive full URLs from the proxy server instead of indexed URLs. For more information about this tag, see The PalmHREFStyle Meta Tag in Chapter 6, "HTML for Web Clipping Applications."

You can also specify that an individual link be downloaded fully by including the `PalmStyle` attribute in the link. For more information, see "The PalmStyle Attribute" in Chapter 6, "HTML for Web Clipping Applications." If you only need a few links to be downloaded fully, you can save on transmission time by using the default, hash coded linking scheme and individually specifying the links that you want fully downloaded.

Using Forms in Web Clipping Applications

You can use forms in your web clipping applications just as you do in other web pages. The only difference in using forms in web clipping applications is that you can use the Palm variables in your forms for special purposes. Listing 4.1 shows an example of a web clipping application that includes a form.

Listing 4.1 Using a Form In a Web Clipping Application

```
<html>
<head>
   <title>Tickets, Inc.</title>
   <meta name="palmcomputingplatform" content="true">
</head>
<body>
<table>
  <tr>
    <td><img src="../images/star.gif">
    <td align="center"><h1>Movies</h1><p><font size="+2">by Tickets, Inc.</
font>
</table><br><br>

<p>
<form method="post" action="http://www.yourwebserver.com/cgi-bin/
moviesform.cgi">
<table cellpadding="1">
   <tr>
      <th align="right">Title:
      <td><input type="text" name="ti" value="Contact" size="15">
   <tr>
      <th align="right">Type:
      <td>
         <select name="type" value="">
            <option selected>Drama
            <option>Action
            <option>Comedy
         </select>
   <tr>
      <th align="right">Date:
      <td><input type="datepicker" name="date" value="8/13/97">
   <tr>
      <th align="right">Time:
      <td><input type="timepicker" name="time" value="8:00 pm">
   <tr>
      <th align="right">Zip:
```

```
      <td><input type="text" name="zip" size="5">  -or- 
      <input type="checkbox" checked name="zip" value="%zipcode">local zip
</table><br><br><br><br>
<p><input type="submit" value="Search">
</form>
</body>
</html>
```

The example in Listing 4.1 is the Movie Query application, which is part of the Sample Code download available from the Palm Developer Support web clipping page at http://www.palmos.com/dev/tech/webclipping/.

The Movie Query application allows you to specify the name and other information about a movie; when you tap the Search button, the application returns the location of the nearest movie theater showing that film. The form used in this application includes a Palm variable and two Palm input types, which are explained in the next section. Figure 4.1 shows how the Movie Query application looks on the handheld screen.

Figure 4.1 The Movie Query sample application

Using Palm Variables and Input Types

You can use the Palm variables shown in Table 4.1 in your web clipping application forms and URL specifications. These variables are described in detail in Palm Variables in Chapter 6, "HTML for Web Clipping Applications."

Table 4.1 Palm form variables

Variable name	Description
`%DeviceID`	Replaced by the proxy server with the device ID of the Palm Powered handheld.
`%Location`	Replaced by the proxy server with location information.
`%WCDevCaps`	Replaced by the device with device-specific information.
`%ZipCode`	Replaced by the proxy server with the 5-digit zip code (or postal code) of the user's current base station.

You can also use the Palm input types shown in Table 4.2 in forms in your web clipping applications.

Table 4.2 Palm form input types

Input type	Description
`DatePicker`	Displays the Palm OS date selector, from which the user can select a date by tapping on fields.
`TimePicker`	Displays the Palm OS time selector, from which the user can select a time by tapping on fields.

The application shown in Listing 4.1 includes a form that makes use of a Palm variable and two Palm input types:

- The `%zipcode` variable. If the user checks the **local zip** checkbox, the Movie Query application sends the user's local zip code to the web site, and the web site returns the location of the theater closest to that zip code that is showing the specified movie.
- The `datepicker` input type. When the user taps the form field with this input type, the Palm OS day selector form displays on the handheld screen, and the user can select a date.

- The `timepicker` input type. When the user taps the form field with this input type, the Palm OS time selector form displays on the handheld screen, and the user can select a time.

Selecting Multiple Form Elements

You can use the `MULTIPLE` attribute with a `SELECT` input type in your forms to allow users to select or scroll multiple items. However, when using this attribute in a web clipping application, you must also specify the `SIZE` attribute, and set the size to a value of at least 2.

For example, Listing 4.2 shows a portion of the form code from the `accomodations` sample application that includes a select input type with the `multiple` attribute.

Listing 4.2 A select form element from the accommodations example program

```
<td align="right" valign="middle"><b>Amenities:</b>
      <td align="left" valign="middle">
         <select name="am" value="" size="3" multiple>
            <option>Business Srvs
            <option>Health Club
            <option>Pool
            <option>Bar
            <option selected>Restaurant
            <option>Travel Srvs
            <option>Any
         </select>
```

The select element in Listing 4.2 specifies a size of 3, which means that the field displays three of the options values in a small window that the user can scroll. Figure 4.2 shows how this application looks on the screen.

Figure 4.2 The accommodations example program

As you can see, the multiple select form element displays a checkbox next to each of the three visible items from which the user can select, and allows the user to scroll through the list and change the visible items.

Actionless Form Tags

One known problem with forms in web clipping applications is that the Viewer does not handle "actionless" form tags well. For example, if you start your form with the following tag:

```
<form>
```

The Viewer does not handle the form well. You can use either of the following techniques to resolve this problem:

- Remove the `<form>` tag, which will also reduce the amount of data that needs to be transmitted. Also note that actionless form tags often result in blank or partial pages displaying on the handheld device.
- Associate an action with the form, even if that action does nothing. For example, if you change the tag to:

  ```
  <form action=bogus.cgi>
  ```

The Viewer displays and activates your form properly.

Additional Information

For additional information about using forms in your web clipping applications, see Chapter 5, "Advanced Web Clipping Actions." You can also find more examples and tips on the Palm Developer web site, at http://www.palmos.com/dev/tech/webclipping/.

Publishing Your Web Clipping Application

Once you create a web clipping application that you want to make available to the public, you can easily do so. Follow these steps:

1. Access the Palm.Net applications area, at http://www.palm.net/apps/. Click the Uploads link.
2. If you do not yet have a Palm.Net developer account, follow the links and fill out the form to create a developer account.
3. Click the Upload New Application link, then read and accept the upload legal agreement that is displayed.
4. Fill out the form with the name, category information, description, and location on your personal computer of your application. For specifics about the fields in the upload form, see Notes on Filling Out the Upload Form.
5. Click the **Save** button to upload your application to Palm.Net.

NOTE: Palm.Net strongly recommends that you upload your web clipping applications in zip format. If you don't already have software to compress files into zip format, you can find freeware, shareware, and buyware versions of such tools on the Internet.

Notes on Filling Out the Upload Form

Table 4.3 provides information to help you to fill in specific fields in the web clipping application upload form on the Palm.Net web site.

Table 4.3 Filling out the application upload form

Form field	Notes
Title	This is the name that will be displayed on Palm.Net for your application.
Version	This is the version number displayed on Palm.Net for your application. This version number should match the version number that you specify in the `PalmLauncherVersion` meta tag of your `index.html` page.
URL	Specify a link to your company's web page, or to a web page that provides additional information about your application. This field is optional.
Category	Select the category in which you want your web clipping application listed on the Palm.Net web site.
File name	Specify the name of the file or archive that you are uploading. **NOTE:** Palm.Net strongly recommends that you provide your file in compressed, zip archive format.
File location	Specify the location on your personal computer of your web clipping application file or archive. You can use the **Browse** button to navigate to the file's location. **NOTE:** The file whose location you specify must have the same name as the file that you specified in File name field.

Table 4.3 Filling out the application upload form *(continued)*

Form field	Notes
Description	A short description of your web clipping application that is displayed in the Palm.Net download area. You must limit your description to two or three short sentences. The Palm.Net administrators may edit your description.
Notes	Any additional information that you want the Palm.Net administrator to know about your application.

5

Advanced Web Clipping Actions

This chapter presents a number of advanced topics related to creating web clipping applications. Read this chapter to learn how to accomplish more sophisticated actions in your web clipping applications. Before reading this chapter, you should be comfortable with the material presented in Chapter 2, "Getting Started with Web Clippings" and Chapter 4, "Creating Web Clipping Applications."

This chapter describes the following topics:

Launching an Application From a Web Clipping

You can use two different tags to launch a Palm OS® application from your web clipping applications:

- Use the `Palm` tag to launch an application. Your web clipping application is terminated, and when the launched application exits, your web clipping application is reinvoked in its initial state.
- Use the `PalmCall` tag to sublaunch an application. Your web clipping application is suspended in the background while the application runs. When the application exits, your web clipping application reactivates in the same state it was in before performing the sublaunch.

NOTE: A launched application can access system globals, but a sublaunched application cannot.

The syntax for launching or sublaunching an application is the same:

```
<a href="Palm:MYAP.appl">textstring</a>
<a href="PalmCall:MYAP.appl">textstring</a
```

where *MYAP* is the four-character creator ID of the application that you want to launch of sublaunch, and *textstring* is the text displayed in the web clipping.

Passing Data Back and Forth

Neither a launched nor a sublaunched application can return data to your web clipping application. However, you can pass data from your web clipping application to the Palm OS application you are calling: use the question mark (?) character to separate the application name from the arguments, and pass each argument in the form `name = value`. For example:

```
<a href="palm:MYAP.appl"?name=Palm, Inc.
city=Santa Clara>Call My App</a>
```

Sending Mail From a Web Clipping

You can send an email message from a web clipping just like you do in standard HTML pages, by using the `mailto:` tag. For example:

```
<a href="mailto:us@company.com">Email us</a>
```

When the Viewer encounters the `mailto` tag, it calls the Exchange Manager to handle sending the email. The Exchange Manager calls the default email application.

New Feature

Registering an Email Application

The standard, default email application on the Palm OS is the Palm `iMessenger` program. Starting with version 4.0 of the Palm OS,

you can register additional email handling applications by calling the Exchange Manager from within a Palm OS application:

```
ExgRegisterDatatype(  CRID, // ID of registering app
                      exgReg                // URL scheme registry
                      "mailto",             // the scheme to associate
                      "Email URL",          // description
                      0 );                  // any flags
```

When you register a new email-handling application, the Exchange Manager makes that application the default handler for email messages. Which means that the newly registered application becomes the application that the Exchange Manager calls when the Viewer encounters the mailto tag.

For more information about the Palm Exchange Manager, see the *Palm OS Programmer's Companion*, vol. II, *Communications* and *Palm OS Programmer's API Reference* books.

Using the Device ID Variable

Your web clipping pages can use the device ID to uniquely identify which Palm Powered™ handheld has generated a specific request. You must use the device ID rather than the source IP address to make this distinction, since the source IP address is not guaranteed to be unique.

Web clipping applications can specify the `%DeviceId` variable as part of a hidden field in a form or as part of an URL specification. The Viewer replaces this variable with a value that is guaranteed to be unique for each device.

Device ID Notes

Note that the Viewer replaces the `%DeviceID` variable with the value `0.0.0` if the web clipping application is running on POSE, the Palm OS Emulator.

All Palm Powered handhelds that support web clipping generate a unique value for the `%DeviceID` variable; however the generated ID is not guaranteed to follow the same guidelines as does the Palm VII™.

For example, a Palm VII device using the Mobitex service includes the Mobitex Access Number as part of the device ID value, but a Palm V™ using the OmniSky service uses the device flash ID instead.

Validating With Device IDs

To validate an address, you need to use the web clipping proxy server source IP address in conjunction with the device ID. If possible, you should also include the username and/or password for validation. For enhanced security, use the HTTPS protocol in the URL specification in your web clipping or web clipping application.

Examples of Using Device IDs

This section presents three examples of using the handheld device ID in a web clipping application.

The first example is a form that includes a hidden field with the device ID:

```
<form method="post" action="https://www.yourserver.com/cgi/
prog.cgi> <input type="hidden" name="did" value="%deviceid">
... </form>
```

The second example passes the device ID as part of an URL specification in a form.

```
<form method="post" action="https://www.yourserver.com/cgi/
prog.cgi?did=%deviceid"> ... </form>
```

The third example shows a device ID in link anchor:

```
<a href="https://www.yourserver.com/cgi/
prog.cgi?did=%deviceid">
```

Downloading Binary Files

If you want to download a file into the user's Palm Powered handheld, you can use the `PalmBinary` attribute in an URL specification in your web clipping application. When the Viewer encounters this attribute in an URL specification, it calls the Exchange Manager to process the download.

New Feature You can only use this feature on version 4.0 or later of the Palm OS. Earlier versions do not support the `PalmBinary` attribute.

The Palm Exchange Manager handles the downloaded file, using criteria such as the file's extension or MIME type to determine which application receives the file.

For example, you could download an application that gets sent to the Palm Applications Launcher with a statement like the following:

```
<a href="http://www.srv.com/files/example.prc" PalmBinary>
```

In the above example, the Exchange Manager knows to send the downloaded file to the Applications Launcher because the file suffix is ".prc".

For more information about the Palm Exchange Manager, see the *Palm OS Programmer's Companion*, vol. II, *Communications* and *Palm OS Programmer's API Reference* books.

Interfacing Palm OS Applications with Web Clippings

This section describes how to interface with the Web Clipping Application Viewer (the Viewer) and web clipping applications from Palm OS programs. It includes the follows sections:

- Using a Palm OS Application to Complete Fields in a Web Clipping Application Form
- Validating Web Clipping Form Data
- Handling Cookies in Web Clipping Applications
- Parsing Data in the Viewer Cache

Most of the solutions in this section involve the use of the INetLib API, which is described in the *Palm OS Programmer's Companion*, vol. II, *Communications* and *Palm OS Programmer's API Reference* books.

Using a Palm OS Application to Complete Fields in a Web Clipping Application Form

You can fill fields in with data in a static web clipping page, but you have to do this without the convenience of an API designed to make it easy. You need to do the following:

- Read the "WCA Encoding Format" chapter of *Palm File Format Specification* to learn how web clippings are formatted.
- Write a Palm OS application that launches the Viewer. You can launch the Viewer to activate a web clipping application, or to access an online URL.
- When you launch the Viewer from your application, you can send parameter data to the URL using the `GET` method. For example:

  ```
  http://www.make_believe_url.com/cgi/
  proc.cgi?name01=val01&name02=val02
  ```

- Note that you cannot use this method to launch a web clipping application and have a specific link in that application automatically activated.

A more robust albeit more complicated option is to write a Palm OS application that uses the INetLib API to perform its own wireless operations; such an application can use the Viewer as a user interface to the Internet.

Validating Web Clipping Form Data

You can use the following steps to have a Palm OS application validate the data in a form that your web clipping application is about to send to a web site:

1. The user fills in the form and taps the **SUBMIT** button to send the form data.
2. Your web clipping application (the PQA) uses either the `PalmCall:` to sublaunch or the `Palm:` tag to launch a Palm OS application. You pass the form data to the application (in the URL that contains that tag. For example:

   ```
   <form action="palm:CRID.appl" method="get">
   <input type="text" name="nm">
   ...
   <input type="submit" value="Submit">
   </form>
   ```

3. The application validates the form and performs actions, as described in the next two subsections, depending on how it was launched.

NOTE: Palm considers launching with the `palm` tag preferable to sublaunching, because it does not generate extra PQA databases on the user's device.

Validating with a Launched Application

If you launched the application (the PRC) with the `palm` tag, then use the following steps:

1. The PRC validates the form data received from the web clipping application.
2. If the validation fails, the PRC displays a modal dialog or form describing the problems.

   The PRC then launches the Viewer with the `sysAppLaunchCmdGoToURL` launch code and passes one of the following values to the Viewer, so that it can redisplay the form page on which the user originally tapped the Submit button:

   - `"file:pqaName.pqa"`
   - `"file:pqaName.pqa/formPageName.html"`
3. If the validation succeeds, the PRC launches the Viewer with the `sysAppLaunchCmdGoToURL` launch code, passing it the validated form data. The Viewer then submits the form and displays the response.

The "Internet and Messaging Applications" chapter of *Palm OS Programmer's Companion*, vol. II, *Communications*, provides information about launching the Viewer from a Palm OS application.

NOTE: The steps described in this section only work with the `Get` form method.

Validating with a Sublaunched Application

If you sublaunched the application (the PRC) with the `palmcall` tag, then use the following steps:

1. The PRC validates the form data received from the web clipping application.
2. If the validation fails, the PRC displays a modal dialog or form describing the problems.

   The PRC then exits, after setting the active form back to that of the PQA page from which it was called.
3. If the validation succeeds, the PRC submits the form using the data passed from the web clipping application.
4. After receiving a response, the PRC either a) renders and displays the response, or b) saves the response as a PQA database and launches the Viewer to display it.

Handling Cookies in Web Clipping Applications

New Feature

Starting with version 4.0 of the Palm OS, the Viewer maintains a cookie jar on each Palm Powered handheld. The server sends cookies to the cookie jar on the handheld device, and the handheld device client sends cookies from the cookie jar to target domains.

The following list describes features of the cookie jar:

- The cookie jar holds up to 50 KB of cookie data.
- Unused cookies are deleted when the cookie jar fills, based on a least-recently-used deletion algorithm.
- Expired cookies are deleted when the Viewer initiates a search of the cookie jar.
- Cookies that expire at the end of a session are automatically deleted when the Viewer exits.

NOTE: When cookies are deleted, all of the cookies for a specific domain are deleted. For example, if cookies coming from `myweb.com` would be deleted at once.

Note that the Web Clipping Prefs panel includes a setting that allows the user to determine if cookies can be sent or received.

Using INetLib to Modify Cookie Settings

You can use the INetLib API to interface with the Viewer's cookie-handling capabilities:

- Call the `InetSettingEnableCookies` function to enable the cookie jar.
- Call the `InetSettingMaxCookieJarSize` function to modify the size of the cookie jar. This setting applies to all applications that use INetLib.
- Call the `InetLibSockHTTPReqSend` function to query the cookie jar.

NOTE: The Viewer's cookie handling is compliant with the Netscape standard.

Redirect Requests

All cookies for a target domain are sent to the web clipping proxy servers, which then sends each cookie to the correct host. Redirect requests are not seen by the client.

Parsing Data in the Viewer Cache

If you want to provide the user with the ability to access a clipping and save it in a Palm OS database, you have two choices:

- You can have the user access the clipping in the history list, and then use the **Copy Page** menu choice to copy the entire clipping. The user can then paste the clipping into an editable field, such as in a memo entry in the Memo Pad application.
- You can create an application that retrieves the clipping from the Viewer's history cache and stores it in a database.

WARNING! The contents of the Viewer cache are neither documented nor supported by Palm, Inc. The cache format is subject to change in future releases of the Palm OS.

If you want to access certain content in one or more clippings, you can write an application that parses the data in the Viewer cache and turns it into text that you can store in a database such as the Memo Pad. If you want to examine the data in the Viewer cache, you need to understand how to map the symbolic constants in the cache into web clipping tags. You can find this information in the "WCA Encoding Format" chapter of *Palm File Format Specification*.

6

HTML for Web Clipping Applications

This chapter provides reference information regarding the specifics of HTML usage for web clipping applications. This chapter presents this information in the following sections:

- About Web Display on Palm Powered Handhelds describes the display and execution environment for HTML display on Palm Powered™ handhelds.
- Meta Tags For Web Clipping Content describes the web clipping-specific meta tags that you can include to control the behavior of web clipping pages.
- Palm Tags for Use in URL Specifications describes the web clipping-specific attributes and strings that you can include in URL specifications and form fields for special meaning.
- Palm Input Types describes the web clipping-specific input types that you can use in form fields.
- Palm Variables describes the web clipping-specific variables that you can include in URL specifications and form fields.
- Palm Tags for Enhanced Productivity describes other HTML extensions that are specific to web clipping.
- Specific Attribute Values for Web Clipping describes specific HTML 3.2 tags, attributes, and features for use in your web clipping pages.
- Unsupported HTML Features describes HTML features that you cannot use in your web clipping pages.
- Validating Your Web Pages tells you how to validate your web pages against the web clipping DTD.

For information on creating your first web clipping application, see Chapter 2, "Getting Started with Web Clippings." For "how-to" information on accomplishing specific tasks in your web clipping applications, see Chapter 4, "Creating Web Clipping Applications."

Appendix B, "Palm HTML Summary," provides a summary description of the HTML that you can use in web clippings, and Appendix C, "Palm OS HTML 3.2 DTD," provides the official Document Type Definition (DTD) understood by the Palm OS® HTML engine.

About Web Display on Palm Powered Handhelds

When the user activates a web clipping application by tapping on it in the Applications Launcher, the Palm OS launches its internal web browsing application, which is known as the Web Clipping Application Viewer (Viewer for short).

The Viewer is the program that executes your web clipping (`.pqa`) applications. It operates as an HTML engine, similarly to the web browsing software that you use on your desktop computer.

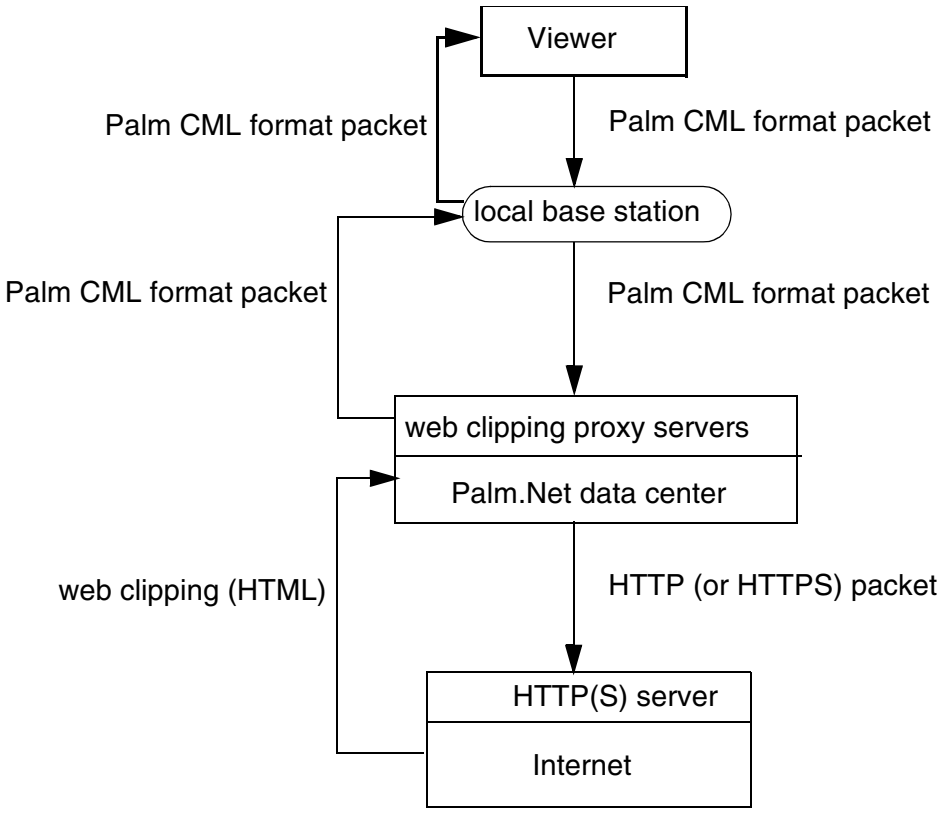
NOTE: Desktop web browsers render HyperText Markup Language (HTML). The Viewer renders Compressed Markup Language (CML), which is a compressed version of HTML developed by Palm, Inc. for use with low-bandwidth connections. You can, however, regard Viewer as a web browser that renders HTML, since any differences are managed by Viewer and the Palm web clipping proxy servers.

When the user activates your application, Viewer displays your top-level (index) page, just as any web browser does. When the user clicks on a link to a remote page, Viewer sends a compressed query to the Palm web clipping proxy servers.

Figure 6.1 shows how Viewer communicates with Internet servers. As you can see, the web clipping scheme depends upon the Palm proxy servers, which mediate between web clipping applications and Internet servers:

- When a Palm Powered handheld sends out a query, the query is relayed in compressed format to the base station nearest to the user.
- The local base station relays the compressed query to the Palm proxy servers, which are located at the Palm.net address.
- The proxy servers convert the compressed query into an HTTP (or HTTPS) packet that is sent to the destination Internet server.
- When the Internet server sends HTML information back to the Palm Powered handheld, the process is reversed: the packet is received by the Palm proxy servers, which convert it into compressed format, which is relayed to the local base station and then back to the handheld.

Figure 6.1 Viewer communication with Internet servers



The web clipping system involves the following communications protocols:

- The Palm OS uses User Datagram Protocol (UDP) to send packets to and receive packets from the base station.
- Communications between the wireless network and the web clipping proxy servers use the TCP/IP protocol.
- Communications between the web clipping proxy servers and Internet servers uses standard Internet protocols: TCP, HTTP, and SSL.

NOTE: The data for HTTPS transactions is encrypted on the handheld device.

The remainder of this chapter describes the enhancements and limitations of using HTML with web clipping applications.

Meta Tags For Web Clipping Content

Palm, Inc. provides several meta tags that you can include in your web clipping code and web clippings, as shown in Table 6.1, which also specifies which versions of the Palm OS support each tag.

IMPORTANT: In the sections that follow, section headings that include "**New!**" describe tags that are only available in version 4.0 or later of the Palm OS. If you use any of these tags in an application, you **must** build the application as a version 2.0 `.pqa` database.

Table 6.1 Palm-specific meta tags

Meta Tag	Description	Usage notes
`HistoryListText`	Specifies the text to display for the page in the history list. See The HistoryListText Meta Tag for details.	For WCA Builder use only. Available in all versions.
`LocalIcon`	Includes images or HTML pages in your application that have not been referenced on local pages. See The LocalIcon Meta Tag for details.	For WCA Builder use only. Available in all versions.
`PalmComputingPlatform`	The web page is web clipping-friendly. See The PalmComputingPlatform Meta Tag for details. **IMPORTANT:** This tag is strongly encouraged for all web clipping pages.	For web clipping pages and WCA Builder use. Available in all versions.
`PalmDoNotCache`	The page is not cached on the device, and type-ahead is disabled on the page. See The PalmDoNotCache Meta Tag for details.	For web clipping pages and WCA Builder use. Available for version 4.0 and later of Palm OS.

Table 6.1 Palm-specific meta tags *(continued)*

Meta Tag	Description	Usage notes
`PalmHREFStyle`	Specifies URLs as full or indexed. See The PalmHREFStyle Meta Tag for details.	For web clipping pages only. Available for version 4.0 and later of Palm OS.
`PalmLauncherName`	The name that is displayed for your application in the Palm OS Applications Launcher. See The PalmLauncherName Meta Tag for details.	For WCA Builder use only. Available in all versions.
`PalmLauncherRevision`	The version number of your web clipping application. See The PalmLauncherRevision Meta Tag for details.	For WCA Builder use only. Available in all versions.
`PalmLargeIconFilename`	The name of the icon file to associate with your web clipping application in the icon view of the Palm OS Applications Launcher. See The PalmLargeIconFilename and PalmSmallIconFilename Meta Tags for details.	For WCA Builder use only. Available in version 4.0 and later of Palm OS.
`PalmPostEncoding`	The encoding to use for posting form data in your application. See The PalmPQABitDepth Meta Tag for details.	For WCA Builder use only. Available in version 4.0 and later of Palm OS.

Table 6.1 Palm-specific meta tags *(continued)*

Meta Tag	Description	Usage notes
`PalmPQABitDepth`	The maximum depth bit-depth for images included in your web clipping application. See The PalmPQABitDepth Meta Tag for details.	For WCA Builder use only. Available in version 4.0 and later of Palm OS.
`PalmPQAVersion`	The minimum version of the Viewer required to run your web clipping application. See The PalmPQAVersion Meta Tag for details.	For WCA Builder use only. Available in version 4.0 and later of Palm OS.
`PalmSmallIconFilename`	The name of the icon file to associate with your web clipping application in the list view of the Palm OS Applications Launcher. See The PalmLargeIconFilename and PalmSmallIconFilename Meta Tags for details.	For WCA Builder use only. Available in version 4.0 and later of Palm OS.

You can also specify the source encoding format for your web clipping pages with the `http-equiv` meta tag, as described in The Source Encoding Meta Tag.

The remainder of this section describes the details of using these meta tags in your web pages.

NOTE: You must place these meta tags in the header section of your web pages, which is the section of the page between the `<head>` and `</head>` tags.

Listing 6.1 shows an example of an HTML page that contains several web clipping meta tags.

Listing 6.1 Web clipping meta tags

```
<html>
<head>
   <title>My Web Clipping Page</title>
   <meta name="palmcomputingplatform" content="true">
   <meta name="historylisttext" content="MyPage &date &time">
</head>
<body>
...
</body>
</html>
```

The HistoryListText Meta Tag

You can include the `HistoryListText` meta tag on web clipping results pages to specify the string that Viewer displays for the page in the history list. The history list is displayed when the user of a web clipping application taps on the History arrow. The format for this meta tag is:

```
<META NAME="HistoryListText" CONTENT="histStr">
```

where *`histStr`* is replaced by the string that you want displayed in the history list. Palm recommends that you include the current time in your history string. You can include the date and/or time using the special `&date` and `&time` variables. For example:

```
<meta name="historylisttext"
  content="MyApp &time &date">
```

Figure 6.2 shows an example of the history list.

Figure 6.2 Text displayed in the history list

If you do not include the `HistoryListText` meta tag on a results page, the Viewer creates a history list string as follows:

- If the results page includes an HTML `TITLE` tag, the history string is the title string, truncated to a length of approximately 12 characters.

- If the results page does not include a `TITLE` tag, the history string is the URL of the page, truncated to a length of approximately 12 characters, with the current time appended.
- Viewer appends the current time, based on the device setting, to the history string it has created.

Palm, Inc. recommends that you include this meta tag in every web clipping results page.

The LocalIcon Meta Tag

The `LocalIcon` meta tag tells the Web Clipping Application Builder tool to include specified local images or HTML pages in your web clipping application. Any images or pages that your pages refer to are automatically included into your application, but you can use the `LocalIcon` tag to force non-referenced pages to also be bundled into your `.pqa` file.

You include these local images or pages so that retrieved web clippings can refer to them. By bundling them into your application, they don't have to be transmitted over a slow connection, which can significantly increase the efficiency of your application.

The format for the `LocalIcon` tag is:

```
<META NAME="LocalIcon" CONTENT="name">
```

where *name* is replaced by the name of the file that you want bundled into your web clipping application.

An example of using this tag is:

```
<META NAME="LocalIcon" CONTENT="MyPhoto.GIF">
```

In the web clipping that is sent back to the handheld, you refer to the included image or file as a location within your `.pqa` file. For example, if your results page wants to display your company's icon, named `myicon.gif`, and your application is named `myapp`, then that page can use a statement similar to the following:

```
<IMG SRC="file:myapp.pqa/myicon.gif">
```

For more information about the Web Clipping Application Builder, see Chapter 7, "Using the Web Clipping Application Builder."

The PalmComputingPlatform Meta Tag

You must include the `PalmComputingPlatform` meta tag on your web pages to designate that they are web clipping-friendly pages. If you want your pages to appear as intended on users' handhelds, you must include this meta tag, which has the following format:

```
<META NAME="PalmComputingPlatform" CONTENT="TRUE">
```

If you do not include this tag on a page, versions of the Viewer earlier than the version in Palm OS 4.0 will not retrieve any images referenced on the page.

Palm, Inc. strongly recommends that you include this meta tag in every page of your web clippings and web clipping application.

New

The PalmDoNotCache Meta Tag

You can include the `PalmDoNotCache` meta tag to prevent storage of the web clipping page on the device. Use this to prevent caching of sensitive data. The format for this tag is:

```
<META NAME="PalmDoNotCache" CONTENT="true">
```

When you include this tag on a results page, the page is not cached, and does not appear in the history list.

NOTE: This tag is only available on devices running version 4.0 or later of the Palm OS. Specifying the `PalmDoNotCache` tag on a web clipping page also disables type-ahead for the page. You can use this to prevent storage of sensitive information such as credit card data.

If you do not specify this tag on a web clipping results page, the page may be cached on the device, type-ahead is enabled for the page, and the page appears in the history list.

New The PalmHREFStyle Meta Tag

You can include the `PalmHREFStyle` meta tag on your web pages to specify that you want the Viewer to receive full URLs from the Palm proxy servers instead of indexed URLs.

The original design of web clipping included the use of an indexed URL scheme: the Palm proxy servers convert each URL into an index number and send that number to the Viewer instead of the full URL. This method, referred to as link hashing, significantly reduces the amount of information that needs to be sent over the airwaves.

Unfortunately, in some circumstances, this method can fail. For example, when the original page reference expires on the server, the indexed URL becomes invalid, and the user may see an error message.

You can include the `PalmHREFStyle` meta tag to tell the Viewer to exchange full URLs with the proxy servers instead of exchanging URL index numbers. The format of this tag is:

```
<META NAME="PalmHREFStyle" CONTENT="value">
```

where *`value`* is either `FULL` or `INDEXED`.

NOTE: This tag only applies when the device is running version 4.0 or later of the Palm OS.

If you do not specify this tag, the Viewer assumes the default value of `INDEXED`.

Note that the `PalmHREFStyle` meta tag sets up the default value for URLs on the page. You can override this setting in an individual link with the `PalmStyle` attribute, as described in The PalmBinary Attribute.

The PalmLauncherName Meta Tag

You can include the `PalmLauncherName` meta tag to specify the name that the Palm OS Applications Launcher displays for your web clipping application. The Web Clipping Application Builder tool (WCA Builder) associates this name with your application. The format for this tag is:

```
<META NAME="PalmLauncherName" CONTENT="Name">
```

where *`Name`* is replaced by the name you want the WCA Builder program to associate with your application. An example of using this tag is:

```
<META NAME="PalmLauncherName"
CONTENT="MyWebClipping">
```

Note that WCA Builder may trim names longer than 8 characters.

You can also specify the launcher name for your web clipping application as a command line option to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

The PalmLauncherRevision Meta Tag

You can include the `PalmLauncherVersion` meta tag to specify a version number for your web clipping application. This is the information string shown for your application when the user chooses the **Info** menu choice in the Application Launcher. The Web Clipping Application Builder tool associates this version with your application. The format for this tag is:

```
<META NAME="PalmLauncherRevision" CONTENT="vStr">
```

where *`vStr`* is replaced by the version string information that you want displayed. This is typically a string with a format such as 2.1. Thus, an example of this meta tag is:

```
<META NAME="PalmLauncherRevision"
CONTENT="2.1">
```

Palm, Inc. recommends that you include this meta tag in the index page of your web clipping application.

You can also specify the version string for your web clipping application as a command line option to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

The PalmPostEncoding Meta Tag

Starting with version 4.0 of the Palm OS, you can include the `PalmPostEncoding` meta tag to specify to the Web Clipping Application Builder tool the encoding that your web clipping application uses for data that it posts via a form. You can use this tag to facilitate converting HTML pages into web clipping applications. The format for this tag is:

```
<META NAME="PalmPostEncoding" CONTENT="Encoding">
```

where *`Encoding`* is replaced by the name of the encoding that you want to use. An example of using this tag is:

```
<META NAME="PalmPostEncoding"
CONTENT="Shift_JIS">
```

You can specify the following values, with punctuation (including the - character) and case ignored:

- `"us-ascci"`
- `"iso-8859-1"`
- `"cp1252"`
- `"shift_jis"`
- `"EUC-JP"`
- `"iso-2022-jp"`

The default encoding used for posted data is `"cp1252."`

NOTE: When the Viewer accesses a page from the Internet, the post data on that page is converted into the character set of the source page. This meta tag has no bearing on that conversion.

You can also specify the posted data encoding type for your web clipping application as a command line option to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

The PalmPQABitDepth Meta Tag

Starting with version 4.0 of the Palm OS, you can include the `PalmPQABitDepth` meta tag to specify the maximum bit depth for images on the web clipping page. The format for this tag is:

```
<META NAME="PalmPQABitDepth" CONTENT="Value">
```

where *`Value`* is replaced by the maximum bit depth that you want the WCA Builder program to use for your application. An example of using this tag is:

```
<META NAME="PalmPQABitDepth" CONTENT="4">
```

You can specify the following values:

- `"1"`
- `"2"`
- `"4"`
- `"8"`
- `"16"`

The default image bit depth is 2. Values of 8 and 16 specify color displays, and other values specify gray-scale displays.

NOTE: When the Viewer accesses a page from the Internet, the bit depth of images on that page is communicated to the handheld device by the proxy server. This meta tag has no bearing on web clipping page contents.

You can also specify the image bit depth for your web clipping application as a command line option to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

The PalmPQAVersion Meta Tag

Starting with version 4.0 of the Palm OS, you can include the `PalmPQAVersion` meta tag to specify the version of the Viewer that your web clipping application requires. If the version available on the user's Palm Powered handheld is lower than the one you specify, your application will not run on that device.

The Web Clipping Application Builder tool (WCA Builder) associates this version information with your application. The format for this tag is:

```
<META NAME="PalmPQAVersion" CONTENT="Value">
```

where *`Value`* is replaced by the number of the Viewer version that your application requires. An example of using this tag is:

```
<META NAME="PalmPQAVersion" CONTENT="2">
```

You can specify the following values:

- `"1"`
- `"2"`

NOTE: This meta tag only applies to web clipping application pages; it is ignored for web clipping pages retrieved from the Internet.

The default version number is 1. Version 2 is the version of the Viewer that was released as part of Palm OS 4.0, which adds a number of features. For more information, see Table 7.1 in Chapter 7, "Using the Web Clipping Application Builder."

You can also specify the required Viewer version number for your web clipping application as a command line option to the WCA

Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

New

The PalmLargeIconFilename and PalmSmallIconFilename Meta Tags

You can include the `PalmLargeIconFilename` and `PalmSmallIconFilename` meta tags to specify the icons that the Palm OS Applications Launcher displays for your web clipping application. The Applications Launcher displays the large icon when the user selects the Icon view, and displays the small icon when the user selects the List view.

The Web Clipping Application Builder tool (WCA Builder) associates these icons with your application. The format for these tags is:

```
<META NAME="PalmLargeIconFilename" CONTENT="filename">
<META NAME="PalmSmallIconFilename" CONTENT="filename">
```

where *`filename`* is replaced by the name of the icon file that you want the WCA Builder program to associate with your application. An example of using these meta tags is:

```
<META NAME="PalmLargeIconFilename"
CONTENT="D:\Icons\MyWebClipBigIcon.gif">

<META NAME="PalmSmallIconFilename"
CONTENT="D:\Icons\MyWebClipSmallIcon.jpg">
```

The icon file must use one of the support icon image formats: JPEG, GIF, or BMP.

You can also specify the launcher icons for your web clipping application as command line options to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

New The Source Encoding Meta Tag

Starting with version 4.0 of the Palm OS, you can include this tag to specify the encoding used for your web clipping pages. The Web Clipping Application Builder tool (WCA Builder) associates this value with your application. The format for this tag is:

```
<META NAME="http-equiv" CONTENT-TYPE="Text/html">;
charset="Encoding">
```

where *Encoding* is replaced by the name of the encoding that you want to use. An example of using this tag is:

```
<META NAME="http-equiv" CONTENT-TYPE="Text/
html">; charset="shift_jis">
```

You can specify the following values, with punctuation (including the - character) and case ignored:

- `"cp1252"`
- `"shift_jis"`

The default encoding used for source data is "`cp1252`."

You can also specify the source encoding type for your web clipping application as a command line option to the WCA Builder program. For more information, see Chapter 7, "Using the Web Clipping Application Builder."

Palm Tags for Use in URL Specifications

Palm, Inc. provides several attributes that you can include in links on your web clipping pages, as shown in Table 6.2.

Table 6.2 Palm-specific attributes and special strings

Attribute or string	Description
`Palm and PalmCall`	Launches or sublaunches a Palm OS application, as described in The Palm and PalmCall Tags.
`PalmBinary`	Allows downloading of a binary file, as described in The PalmBinary Attribute
`PalmStyle`	Specifies exchange of full or indexed URL for a specific link. For more information, see The PalmStyle Attribute

You can only include these tags in limited locations in your web clipping pages. You can include them:

- In anchor tags with `HREF` attributes as part of the URL string.
- In form action attribute values as part of the URL string.
- In form field values as the "value" part of a name/value pair. If the `GET` method is used for the form, the name/value pairs are appended to the action attribute URL value.

The remainder of this section describes the details of using these tags in your web pages.

NOTE: These tag names are not case sensitive.

The Palm and PalmCall Tags

You can use these tags to launch or sublaunch a Palm OS application from your web clipping application. You need to know the application's creator ID to use these tags. Use the `Palm` tag inside of a link to launch an application, use the `PalmCall` tag to sublaunch an application. The syntax for both tags is identical:

```
<a href="Palm:MYAP.appl">textstring</a>
<a href="PalmCall:MYAP.appl">textstring</a
```

where *MYAP* is the four-character creator ID of the application that you want to launch of sublaunch, and *textstring* is the text displayed in the web clipping.

NOTE: You must supply the four-character creator ID of the application, **not** the application name. And you must use all capital letters for the creator ID.

You can pass name/value pair arguments to the application that you are launching: use the question mark (?) character to separate the application name from the arguments, and pass each argument in the form `name = value`. For example:

```
<a href="palm:MYAP.appl"?name=Palm, Inc.
city=Santa Clara>Call My App</a>
```

Launching Versus Sublaunching

When a sublaunched application exits, control is returned to the calling application that was left in the background. In the case of using `PalmCall` to sublaunch an application from a web clipping, this means that when the sublaunch ends, the web clipping form is in the same state that it was in when you made the call.

When a launched application exits, control is returned to the web clipping, but the form that used the `Palm` tag to launch the application is redisplayed in its initial state.

Note that you cannot return data to the web clipping form from a launched or sublaunched application.

New The PalmBinary Attribute

Starting with version 4.0 of the Palm OS, you can use the `PalmBinary` attribute in an URL specification to specify a binary file for download into the Palm Powered handheld. The Palm Exchange Manager handles the downloaded file, using criteria such

as the file's extension or MIME type to determine which application receives the file.

For example, you could download an application that gets sent to the Palm Application Launcher with a statement like the following:

```
<a href="http://www.srv.com/files/example.prc" PalmBinary>
```

For more information about the Palm Exchange Manager, see the *Palm OS Programmer's Companion*, vol. II, *Communications* and *Palm OS Programmer's API Reference* books.

New The PalmStyle Attribute

Starting with version 4.0 of the Palm OS, you can use the `PalmStyle` attribute in an URL specification to define whether the Palm proxy server sends the full URL or the URL index to the handheld. Including this attribute in the URL overrides the default value for the page, which is set with The PalmHREFStyle Meta Tag.

The following are two examples of using the `PalmStyle` attribute:

```
<a href="http://www.palmos.com/" PalmStyle="full"> The Full
Link</a>

<a href="http://www.palmos.com/" PalmStyle="index"> The
Indexed Link</a>
```

The value of this attribute can be either `full` or `index`. For more information, see The PalmHREFStyle Meta Tag.

Palm Input Types

Palm, Inc. provides several input types that you can use in forms in your web clipping applications, as shown in Table 6.3.

Table 6.3 Palm-specific input types

Input type	Description
`DatePicker`	Displays the Palm OS date selector. For more information, see The DatePicker Input Type.
`TimePicker`	Displays the Palm OS time selector. For more information, see The TimePicker Input Type.

The DatePicker Input Type

You can use the `DatePicker` input type to insert the current date or assign a value to the date. The syntax for using the `DatePicker` input type is:

```
<input type="datepicker" name="date" value="value">
```

where *`date`* is replaced by the form field name you supply, and *`value`* is optionally replaced by a date that you supply.

If you supply a *`value`*, then that is the value of the field. If you do not supply a *`value`*, then the Viewer displays the Palm OS date selector, from which the user can select a date by tapping on fields.

This form field returns its value in the format *YYYY-MM-DD*. Note that the Viewer displays the date value in *MM/DD/YY* format unless the user preferences specify otherwise.

The TimePicker Input Type

You can use the `TimePicker` input type to insert the current time or assign a time value to a form field. The syntax for using the `TimePicker` input type is:

```
<input type="timepicker" name="time" value="value">
```

where *`time`* is replaced by the form field name you supply, and *`value`* is optionally replaced by a time that you supply.

If you supply a *`value`*, then that is the value of the field. If you do not supply a *`value`*, then the Viewer displays the Palm OS time selector, from which the user can select a time by tapping on fields.

This form field returns its value in the format *HH-MM* twenty-four hour clock format. Note that the Viewer displays the time value in *HH:MM:AM/PM* format unless the user preferences specify otherwise.

Palm Variables

Palm, Inc. provides several variables that you can include in links and form fields on your web clipping pages, as shown in Table 6.4.

Table 6.4 Palm-specific variables

Attribute or string	Description
`%DeviceID`	Replaced by the proxy server with the device ID of the Palm Powered handheld. For more information, see The %DeviceID Variable
`%Location`	Replaced by the proxy server with location information. For more information, see The %Location Variable
`%WCDevCaps`	Replaced by the device with device-specific information. For more information, see The %WCDevCaps Variable
`%ZipCode`	Replaced by the proxy server with the 5-digit zip code (or postal code) of the user's current base station. For more information, see The %Zipcode Variable

You can only include these variables in limited locations in your web clipping pages. You can include them:

- In anchor tags with `HREF` attributes as part of the URL string.
- In form action attribute values as part of the URL string.
- In form field values as the "value" part of a name/value pair. If the `GET` method is used for the form, the name/value pairs are appended to the action attribute URL value.

The remainder of this section describes the details of using these variables in your web pages.

NOTE: The Palm variable names are not case sensitive.

The %DeviceID Variable

You can use the `%DeviceID` variable to retrieve the unique device ID of the user's Palm Powered handheld. Use a hidden field with the following syntax to use the `%DeviceID` variable in a form:

```
<input type="hidden" value="%deviceid" name="id">
```

where *id* is replaced by the form field name you supply.

The value of the form field is the device ID, in the following format:

```
validationCode.deviceId
```

where the Palm proxy server replaces *validationCode* with an integer value, as shown in Table 6.5, and *deviceId* is replaced by the unique user/organizer ID that was created when the device on which the Viewer is running was activated.

The possible validation values are shown in Table 6.5.

Table 6.5 Validation values of the `%DeviceId` variable

Value	Description
-1	The device is not a Palm Powered handheld that supports web clipping.
0	The proxy server cannot determine if the device is a Palm Powered handheld that supports web clipping.
1	The Palm web clipping proxy server recognizes the device ID as valid.

If the Viewer is running in the Palm OS Emulator, the value of `%DeviceId` is "0.0.0".

You can use this variable in forms to differentiate among different handhelds. You can also use `%DeviceID` in an URL specification, as shown in the following example:

```
<form method="get" action="http://
www.company.com/cgi?id=%deviceId">
```

WARNING! The format of the `%DeviceId` value may change in the future. Your code should treat this value as a black box.

The %Location Variable

Starting with version 4.0 of the Palm OS, can use the `%Location` variable to insert network data from the user's carrier network in URL specifications.

The `%Location` string has the following format:

```
%Location:<code> (<separator>? <code>)* "."
```

where the `<code>` is one of the codes shown in Table 6.6, and each separator is a separator character such as the semicolon (;), colon (,), or pound (#). The following are valid examples of a location string specification:

```
%location:z;c.
%location:z#c.
%location:c#Y.
```

NOTE: The separators between codes are optional.

To use the `%Location` variable in a form field, use a field with the following syntax:

```
<input type="hidden" name="loc" value="%Location">
```

where *loc* is replaced by the form field name you supply. Note that, although the syntax specification shows a hidden field, you can use this variable in any input field, hidden or not.

The value of *%Location* is filled in when a form is submitted, or when the Viewer follows the URL specification that includes the variable.

You can also use the %Location variable in an URL specification such as the following:

```
<a href="http://svr.com/proc.cgi?loc=
%location:z;c.">My Location</a>
```

Table 6.6 summarizes the codes you can include in location variable specifications. Note that the codes are case sensitive.

Table 6.6 Location information codes

Code	Information type	Notes
c	City name	The name of the city.
G	Global Positioning System	The longitude and latitude for the location.
l	State name	Also used for the name of the large political division in countries other than the U.S.A.
L	State code	Also used for the code of the large political division in countries other than the U.S.A.
O	County code	Also used for the code of the small political division in countries other than the U.S.A.
o	County name	Also used for the name of the small political division in countries other than the U.S.A.
R	Raw location data	The raw location data, in hexadecimal.
y	Country name	The name of the country.

Table 6.6 Location information codes *(continued)*

Code	Information type	Notes
`Y`	Country code	The code is based on Codes for the Representation of Names of Countries, ISO 3166-1993 (E), from the American National Standards Institute, Inc.
`z`	Zip code	Also used for postal code in countries other than the U.S.A.

New The %WCDevCaps Variable

Starting with version 4.0 of the Palm OS, you can use the `%WCDevCaps` variable to insert web clipping device capability information into URL specifications.

To use the `%WCDevCaps` variable in a form field, use a field with the following syntax:

```
<input type="hidden" name="cap" value="%WCDevCaps">
```

where *`cap`* is replaced by the form field name you supply. Note that, although the syntax specification shows a hidden field, you can use this variable in any input field, hidden or not.

The value of *`%WCDevCaps`* is filled in when a form is submitted. The value is a hexadecimal string of undefined length that specifies device information, as described in Table 6.7. If the Palm Powered handheld does not support this feature, `%WCDevCaps` is ignored in the URL.

NOTE: If you receive the string `%WCDevCaps` instead of a hexadecimal string, you know that the device is running a version of the Palm OS earlier than version 4.0, and thus you cannot take advantage of the new features described in this book.

Table 6.7 lists the capabilities currently reported in the hex string. Note that additional reporting will be added to this string in the future.

Table 6.7 Web clipping device capability string contents

Capability	Bits used	Description
Screen	0 = 1 bit monochrome 1 = 2 bit monochrome 2 = 4 bit monochrome 3 = 8 bit color 4 = 16 bit color 5 and 6 are reserved	The bit depth supported by the handheld device. Note that this setting does not determine the bit depth of what gets downloaded.
Lz77 Available	7	Defines if the Lz77 library is available in the installed version of the Palm OS.
Communications bandwidth	8 to 11	Values: 0 = unknown 1 = Mobitex Other values are currently undefined.
OS free memory	12 to 15	Values: 0 = 2^16 (< 64 KB) 1 = 2^17 2 = 2^18 ... 14 = 2^30 15 = 2^31 For example, if the value is 2, then: 128 KB <= size <= 256 KB

Table 6.7 Web clipping device capability string contents *(continued)*

Capability	Bits used	Description
Palm OS version	16 to 20	Values: 0 = unknown 1 = 3.2 2 = 3.3 3 = 3.5 4 = 4.0 others currently undefined
OS heap size	21 to 23	Values: 0 = > 256K 1 = 64K 2 = 96K 3 = 128K 4 = 256K others currently undefined

The %Zipcode Variable

You can include the `%Zipcode` variable in your web clipping pages to specify the 5-digit ZIP code or postal code of the base station with which the user is currently communicating. The following is the syntax for using the `%Zipcode` variable in a form.

```
<input type="hidden" name="zip" value=%zipcode>
```

where *zip* is replaced by the form field name you supply.

The value of *%zipcode* is filled in when a form is submitted, or when the Viewer follows the URL specification that includes the variable.

If the ZIP code request originates from a base station whose ZIP code has not been entered or from an emulated Palm Powered handheld, a value of `"00000"` is assigned to the variable.

You can also use `%zipcode` variable in an URL specification, as shown in the following example:

```
<form method="get" action="http://
www.company.com/cgi?zip=%zipcode>
```

NOTE: Some base stations overlap ZIP codes, so accuracy is not guaranteed.

Palm Tags for Enhanced Productivity

You can use other web clipping-specific extensions to enhance the productivity of your web clipping pages, as shown in Table 6.8.

Table 6.8 Other web clipping extensions to HTML

Tag	Description
`Button`	Creates a labelled hyperlink button.
`SmallScreenIgnore`	Specifies web page content that is ignored by web clipping.

The Button Tag

You can use the button tag to create a button with a label and a hyperlink. The button can link to local or remote pages or images.

Use the following syntax to link to local page in your application:

```
<a href="page.html" button>label</a>
```

where *page* is the name of the HTML page, and *label* is the text label that you want displayed in the button.

Use the following syntax to refer to a remote page:

```
<a href="http://www.domain/page.html" button> label </a>
```

where *domain* is the domain name, *page* is the name of the page in that domain, and *label* is the text label that you want displayed in the button.

The SmallScreenIgnore Tag

You can use this tag to specify that certain content on a page is to be ignored for web clipping. If the tag is encountered in a page used to

build your application, the Web Clipping Application Builder tool ignores the content. If the tag is encountered in a results page, the proxy server does not send the content.

You can use this tag to create pages that can be viewed with the Viewer and on desktop web browsers.

The format of this tag is:

```
<SMALLSCREENIGNORE>
...
<!-- HTML Content -->
...
</SMALLSCREENIGNORE>
```

Any content between the `<SMALLSCREENIGNORE>` start tag and the `</SMALLSCREENIGNORE>` end tag is ignored for web clipping.

Specific Attribute Values for Web Clipping

This section describes specific values and limitations for HTML tag usage with web clipping. Table 6.9 summarizes the HTML tags for which there are Palm-related specifics, and the following sections provide more information about specific HTML elements:

- Fonts on Web Clipping Pages
- Colors on Web Clipping Pages
- Images on Web Clipping Pages

Table 6.9 HTML element specifics for web clippings

Tag	Description	Notes
`<BODY>`	`BGCOLOR` attribute	See Colors on Web Clipping Pages.
`<BODY>`	`TEXT` attribute	See Fonts on Web Clipping Pages.
`<FONT>`		See Fonts on Web Clipping Pages.

Table 6.9 HTML element specifics for web clippings *(continued)*

Tag	Description	Notes
<FORM>	METHOD, ACTION, and ENCTYPE attributes	The Viewer supports the POST and GET methods. The Viewer also supports the default ENCTYPE value of application/x-www-form-url-encoded.
<H1> ... <H6>	ALIGN	See Fonts on Web Clipping Pages.
<IMG>		See Images on Web Clipping Pages.
<INPUT>	TYPE, NAME, VALUE, CHECKED, SIZE, SRC, ALIGN, and MAXLENGTH attributes	You can specify the following values for TYPE: • PASSWORD • CHECKBOX • RADIO • SUBMIT • RESET • HIDDEN • TIMEPICKER • DATEPICKER

Table 6.9 HTML element specifics for web clippings *(continued)*

Tag	Description	Notes
<TABLE>		The Viewer displays tables without a border unless you specify the BORDER attribute. Tables are sized to "best-fit" unless you specify the WIDTH attribute. The maximum width of a table is 153 pixels. The Viewer does not support nested tables; if your clipping includes them, the nested tables are rendered as text: table formatting is removed and the content within the table, which includes the text and any other HTML elements, is rendered.
<TEXT>	NAME, VALUE, SIZE, and MAXLENGTH attributes	You can use the NAME, VALUE, SIZE, and MAXLENGTH tags in forms with the syntax such as the following: <INPUT TYPE=TEXT SIZE=8>
<UL>		The Viewer displays a maximum of five list levels. Nested lists are supported to the width of the screen.

Fonts on Web Clipping Pages

The Web Clipping Application Viewer (the Viewer) supports only one typeface for display web clipping content: the Palm TD typeface, which has been optimized for displayed on small screens.

You can use standard HTML tags to specify font face changes, including boldface and italics, and you can use the following HTML tags to change font sizes:

- <SMALL>

- `<BIG>`
- `<FONT SIZE=`*n*`>`
- `<BASEFONT SIZE=`*n*`>`

Note that TD 9, which is the normal font for plain text, corresponds to `<FONT SIZE=3>`.

IMPORTANT: Remember that the content area for web clippings is 153x144 pixels. If you are using TD 9, you can only fit 11 lines of text in the content area.

Colors on Web Clipping Pages

The Viewer supports several different levels of color display, depending on the type of Palm Powered handheld that the user has:

- 2-bit grayscale
- 4-bit grayscale
- 8-bit color
- 16-bit color

On 2-bit grayscale systems, colors are mapped into the colors shown in Table 6.10.

Table 6.10 Web clipping colors for grayscale systems

Color name	Color value
Black	`#000000`
Silver	`#C0C0C0`
Gray	`#808080`
White	`#FFFFFF`

On 4-bit grayscale systems, colors are mapped to the appropriate gray-scale values, which include the values `#000000`, `#101010`, `#202020`, `#303030`, and so on, up to `#F0F0F0`.

On color systems, you can use the NetScape web-safe color palette.

NOTE: Handhelds running version 4.0 or later of the Palm OS send device capability information to the proxy server, which allows the proxy server to send images with the correct bit depth back to the device.

Images on Web Clipping Pages

The following list briefly describes several issues that you need to consider when including images in your web clipping pages. For more detailed information see "Using Images in Web Clippings" in Chapter 4.

- You must include the `PalmComputingPlatform` meta tag on any web clipping page that includes an image.

NOTE: Palm recommends that you include this meta tag on every web clipping page, but you **must** include it if you want an image to correctly display on the page.

- Every image that you reference in your web clipping application that is stored on your personal computer is copied into your `.pqa` file.
- You can include image files in your web clipping application (`.pqa`) file even if the images are not referenced in your web clipping application. You can do this to store images that are referenced by online pages, which can significantly reduce the amount of data that needs to be transmitted.
- The maximum displayble width of an image on the Palm VII™ and similar Palm Powered handhelds is 153 pixels.

New Feature

On devices running version 4.0 or later of the Palm OS, wider images can be scrolled if the containing page includes the `PalmComputingPlatform` meta tag. On earlier versions of the Palm OS, wider images do not display.

- You can use most of the standard image attributes, including: `ALIGN`, `ALT`, `BORDER`, `HSPACE`, `VSPACE`, `HEIGHT`, and `WIDTH`. You cannot use the `ISMAP` or `USEMAP` attributes.

- Images may not align correctly if the containing page does not include the `PalmComputingPlatform` meta tag.
- The Viewer treats image borders differently than many browsers:
  - The default border value is 0.
  - To display a border, you must specify an even number value.
  - Borders are always drawn as 1 pixel wide.
  - The Viewer does not automatically draw a border around an image that is part of an anchor tag.

Unsupported HTML Features

The Viewer supports most of the HTML 3.2 standard. Table 6.11 summarizes the features that are not supported by the Viewer.

Table 6.11 Unsupported HTML Features

HTML Feature	Notes

Specific tags	`<VSPACE>`
	`<SUB>`
	`<SUP>`
	`<LINK>`
	`<ISINDEX>`

Cascading style sheets	
Frames	
Image maps	
Animated GIFs	
Java applets	
JavaScript	
Layers	

Table 6.11 Unsupported HTML Features *(continued)*

HTML Feature	Notes
Named typefaces	For more information about the fonts you can use, see Fonts on Web Clipping Pages.
Nested tables	Rendered as plain text.

Note that the Palm proxy servers strip out unsupported HTML code, which means that if you attempt to view a page that includes such features, the page still displays; however, the appearance of such pages is likely to be unpleasing.

Validating Your Web Pages

The World Wide Web Consortium (W3C) web site provides a program that validates a set of HTML pages against a Document Type Definition (DTD). The HTML validator program is located at the following URL:

http://validator.w3.org/

To use this program to validate that your pages conform to the Palm HTML 3.2 subset, you must tell the validation program where to find the Palm DTD. To do so, add the following `DOCTYPE` statement before the opening `<HTML>` tag in your top-level page:

```
<!DOCTYPE HTML PUBLIC "-//POS//DTD WCA HTML
1.1//EN" "http://www.palm.com/dev/webclipping-
html-dtd-11.dtd">
```

The validator program works with an online location or with files on your personal computer.

7

Using the Web Clipping Application Builder

This chapter describes how to use the Web Clipping Application Builder (*WCA Builder*) program to create a web clipping application that you can execute on Palm Powered™ handhelds. You use WCA Builder on your desktop computer to generate your application from a set of HTML page components, and then install the built application on the handheld.

This chapter contains the following sections:

- About the WCA Builder Program provides general information about the program and describes how to prepare your files before using the WCA Builder program.
- Using the WCA Builder User Interface describes using the user interface of the WCA Builder program to create a web clipping application.
- Using the WCA Builder Command-Line Interface describes how to use the command line interface to WCA Builder.
- Using the Image Checker Program describes how to use the Image Checker program, which allows you to preview and modify images so that you can see how they look on the handheld screen. The Image Checker is built into the Macintosh version of WCA Builder, and is a separate program in other versions.

About the WCA Builder Program

This section describes how to use the WCA Builder program to convert your web clipping application pages into a web clipping application (`.pqa`) file that runs on Palm Powered handhelds.

This chapter describes the user interface to the Windows version of the Web Clipping Application Builder (WCA Builder) program. Other versions provide the same interface capabilities, though the interface might be slightly different.

Preparing Your Web Clipping Application Files

Before you can build your web clipping application, you need to prepare your application's files according to the following checklist:

- Your web clipping application pages should conform to the Palm HTML 3.2 subset, which is discussed in Chapter 6, "HTML for Web Clipping Applications," and summarized in Appendix B, "Palm HTML Summary."
- Make sure that you know which page is your top-level page. This is typically named `index.html`, but you can use any name that you wish.
- Make sure all pages and graphics have unique names. You cannot rely on file extensions to differentiate between names with WCA Builder. For example, WCA Builder sees the files `Myapp.html` and `Myapp.gif` as having the same name.
- Your HTML files can compile with either `.htm` or `.html` extensions.
- Store all of your web clipping application files, including image files, under one root directory. You can include subdirectories.

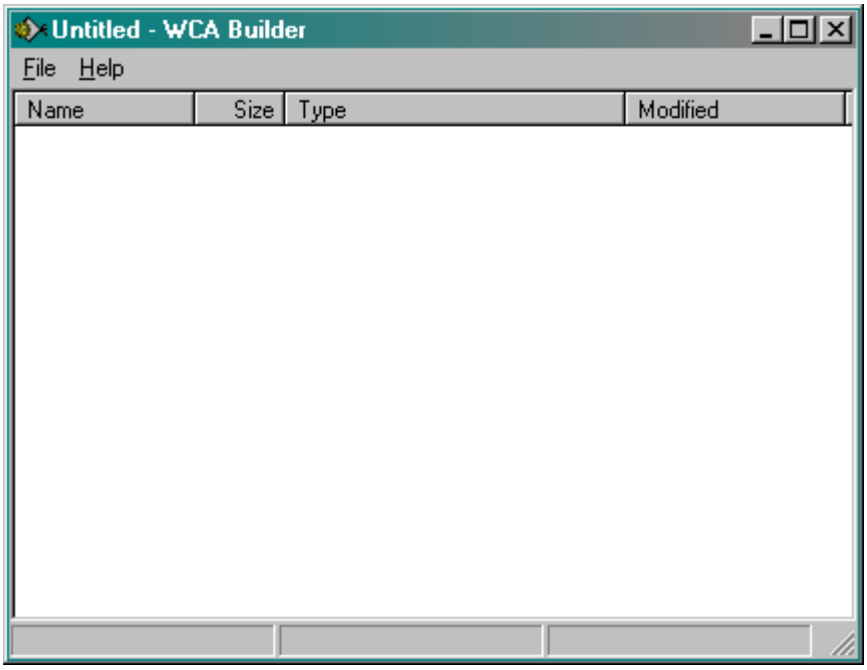
NOTE: Although you can store your files in subdirectories, any links that you make to other local files within your web clipping application assume that the file is in the same directory. For more information, see Linking to Local Files in Chapter 4, "Creating Web Clipping Applications."

- Prepare your graphic images for use with your web clipping application. For more information, see Using Images in Web Clippings in Chapter 4, "Creating Web Clipping Applications."
- Use the Image Checker, as described in Using the Image Checker Program, to verify that your images display properly on the handheld screen.

Using the WCA Builder User Interface

When you first start the WCA Builder application, it displays a browsing window, as shown in Figure 7.1.

Figure 7.1 WCA Builder window



This window displays the files that are part of the web clipping application you are building. When you open your top-level HTML page in WCA Builder, the program automatically opens all of the files that are part of the web and displays them in this window.

The WCA Builder program is simple to use. To build your web clipping application, follow these steps, which are described in this section.:

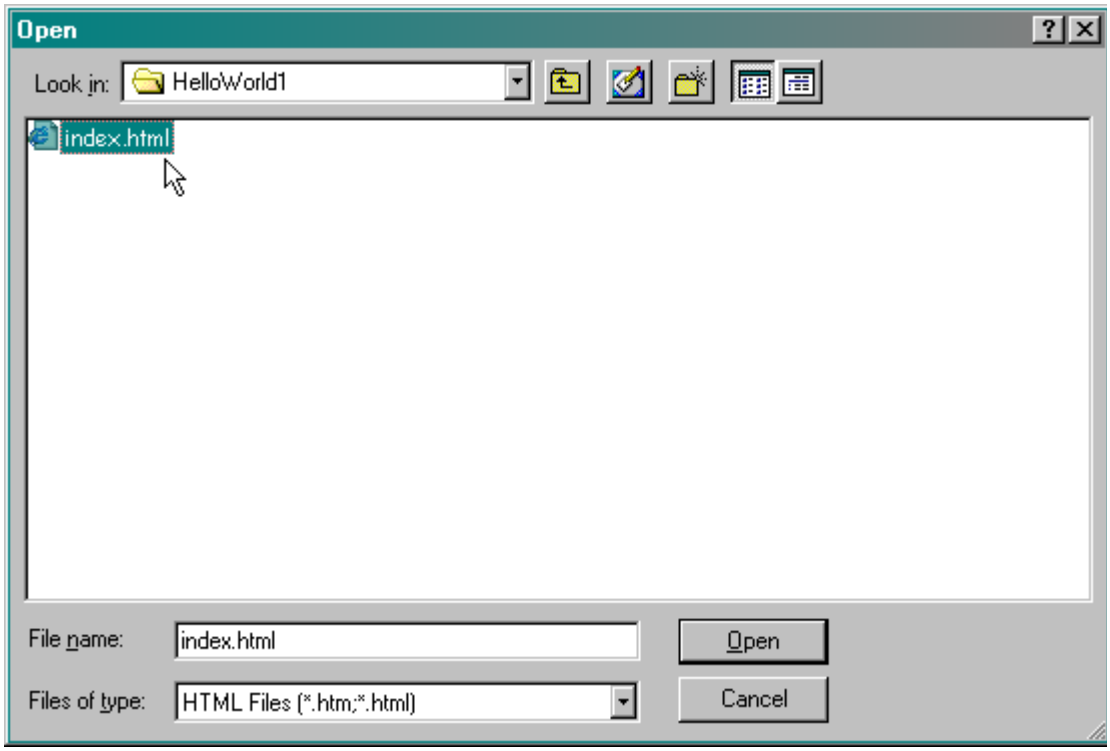
1. Open your top-level web page. Select the **Open Index** command from the File menu, and open the top-level page of your HTML web. WCA Builder opens that page and any other pages or images associated with it.
2. Select the **Build PQA** command from the File menu. WCA Builder displays the Build PQA dialog box.

3. Specify the name and destination for your web clipping application `.pqa` file.
4. Specify options for your web clipping application, including the encoding and image depth to use, and customized icons for display in the Applications Launcher.
5. Click the Build button in the dialog box. WCA Builder builds your web clipping application and saves it in the location that you specified.

1. Open Your Top-Level Web Page

Your first step is to have WCA Builder open your web clipping application files. Select the **Open Index** command from the File menu. WCA Builder displays a standard file opening dialog box that you can use to navigate to the location of your top-level page, as shown in Figure 7.2.

Figure 7.2 Opening your top-level page in WCA Builder



When you open your top-level file, the WCA Builder program scans the file and any files referenced by it, opening each referenced file. After WCA Builder has finishing scanning and opening the files, it displays the name and size of each file that is part of your application in its browsing window.

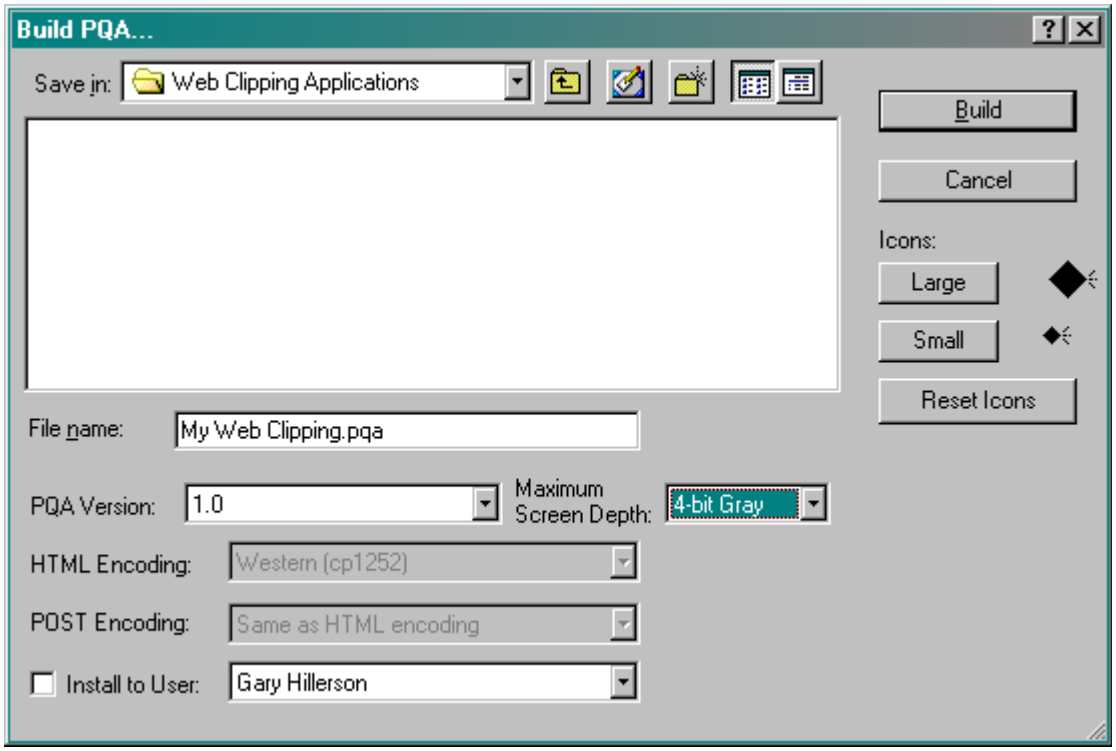
If you make modifications to any of the files while you have WCA Builder open, you can have WCA Builder rescan your web by selecting the **Rescan HTML** command from the File menu.

If you do not see a file that you expected to see listed, then the file is not referenced from anywhere in your application. You need to check that you have included the correct links and META tags in your HTML files. If you make any changes, be sure to use the **Rescan HTML** command before building your application.

2. Select the Build PQA Command

To set up the options for and build your web clipping application, select the **Build PQA** command from the File menu. WCA Builder displays the Build PQA dialog box, which is shown in Figure 7.3.

Figure 7.3 The Build PQA dialog box



As you can see, the top portion of this dialog box shows the destination directory and provides familiar buttons for navigating to other directories. The following sections describe the other features of the Build PQA dialog box.

3. Specify the Name and Destination for Your Application

You need to enter the name of the `.pqa` file in which your web clipping application is saved in the **File name** field of the Build PQA dialog box. This is the name of the saved file and is also the name displayed for your web clipping application in the Palm OS® Applications Launcher.

NOTE: In the MacOS version of the Web Clipping Application Builder, you can specify different values in the dialog box for the `.pqa` file name and the program name that is displayed in the Applications Launcher. In other versions of the program, you can specify a different Applications Launcher name as a command line option, as described in Using the WCA Builder Command-Line Interface, or in a META tag that you specify in your top-level web page, as described in Chapter 6, "HTML for Web Clipping Applications.".

When you build your web clipping application, WCA Builder saves the application in the name that you specify in this field, in the directory that is displayed at the top of the Build PQA dialog box.

4. Specify Options for Your Application

You can use fields in the Build PQA dialog box to specify a number of options for your web clipping application. This section describes how to use these options:

- Specifying The PQA Version Number describes how to specify the version number of the Web Clipping Application Viewer (the Viewer) that your web clipping application requires.

- Setting the Image Depth describes how to specify the maximum bit depth of images in your application.
- Specifying the Encoding Format describes how to change the HTML and form posting formats used in your application.
- Automatically Installing Your Web Clipping Application describes how to have the WCA Builder program automatically set up your web clipping application to be installed on your handheld when you next perform a HotSync® operation.
- Changing the Launcher Icons describes how to customize the icons that the Palm OS Applications Launcher displays for your web clipping application.

Specifying The PQA Version Number

Use the **PQA Version** field of the Build PQA dialog box to specify which version of the Palm OS your application requires, and which web clipping features you can use. At the time of this writing, there are two versions available:

- Version 1.0 (the default) is the original version of the Viewer, and is the version available on Palm Powered handhelds running versions of the Palm OS earlier than version 4.0. This version includes all of the basic web clipping functionality.
- Version 2.0 is the version of the Viewer released with Palm OS version 4.0. This version adds several important features, which are summarized in Table 7.1. If your application includes any of these features, you **must** build it as a version 2.0 `.pqa`.

Table 7.1 Features added to version 2.0 web clippings

Feature	Description
Palm-specific meta tags	Version 2.0 of the Viewer adds supports for a number of Palm-specific meta tags. For more information, see Meta Tags For Web Clipping Content and Palm Tags for Use in URL Specifications in Chapter 6, "HTML for Web Clipping Applications."
Palm-specific variables	Version 2.0 of the Viewer adds supports for a number of Palm-specific variables, including the `%WCDevCaps` variable. For more information, see Palm Variables in Chapter 6, "HTML for Web Clipping Applications."
Color	Version 2.0 of the Viewer adds support for color text, images, and backgrounds on web pages.
2-byte characters	Version 2.0 of the Viewer supports 2-byte characters.
Image depth	The maximum image depth for web clipping pages was increased from 8 bits to 16 bits.
Lz77 compression	Palm, Inc. implemented Lz77 compression for version 2.0 of the Viewer, which provides improved compression of web clipping data, especially for double-byte data.

Table 7.1 Features added to version 2.0 web clippings *(continued)*

Feature	Description
Cookie support	Version 2.0 of the Viewer allows for storing cookies on the user's device.
Binary upload w/o pre- or post-processing	Version 2.0 of the Viewer eliminates the need for you to pre-process binary data that you are uploading, and the need for the server to post-process that same data.

WARNING! If you specify that your `.pqa` depends on version 2.0, WCA Builder builds it in such a way that it cannot run with earlier versions of the Viewer. Users who attempt to run these applications on versions of Palm OS earlier than 4.0 will receive an error message.

Setting the Image Depth

You can specify a maximum bit-depth for images that are included in your web clipping application. When your application runs on a user's Palm Powered handheld, the Viewer displays images in your application using a bit depth determined by two factors:

- The bit depth of the images in the application, and/or
- The maximum bit depth of the device

If an image is encoded to a greater bit depth, then devices that can handle that depth display the image; however, if the device cannot handle the bit depth of the encoded image, the image displays at a lower bit depth (and resolution). You can make all of your images display similarly across device types by deliberately choosing a lower bit depth, such as 2-bit or 4-bit images.

If you specify a larger bit-depth for your images, they will look better on users' devices that have enhanced grayscale or color displays. However, larger bit depth images increase the size of your `.pqa` file, and can result in slower execution.

Table 7.2 shows the values that you can specify for the maximum bit depth of images in your application:

Table 7.2 Web clipping application image depth values

Value	Descriptions
1-bit B&W	Black and white displays
2-bit Gray	Simple grayscale displays
4-bit Gray	Advanced grayscale displays
8-bit Color	Early color displays
16-bit Color	Advanced color displays

Specifying the Encoding Format

If you have specified that your web clipping application is being built for version 2.0 of the Viewer, you can also specify that it get built using different encodings:

- You can specify that the HTML uses the Western (cp1252) encoding or the Japanese (shift_jis) encoding.
- You can specify that forms in your web clipping application post their data using one of the following encodings:
  - 7-bit (us-ascii)
  - Western (iso-8859-1)
  - Western (cp 1252)
  - Japanese (shift-jis)
  - Japanese (EUC-JP)
  - Japanese (iso-2022-jp)

Automatically Installing Your Web Clipping Application

If you want to install your built application to a Palm Powered handheld, you can save yourself a step by having WCA Builder store a copy of the built `.pqa` file in the Install folder of the user name that you specify. This is the same action that occurs when you use the Palm Desktop Software Install tool and tell it to install your

`.pqa` file. The net result is that your `.pqa` file is installed on the device when that user next performs a HotSync operation.

To have WCA Builder add your `.pqa` file to the list of applications to be installed during the next HotSync operation, you must both check the **Install to User** checkbox and select the name of the user for whom the install is to occur.

Users can also install .pqa files over a wireless connection with the MyPalm portal.

Changing the Launcher Icons

The Applications Launcher on Palm Powered handhelds displays an icon for each application on the device. WCA Builder allows you to specify the icons that you want associated with your web clipping application in the Applications Launcher, which uses two different icons:

- A small icon that is displayed for users who have chosen List view in the Applications Launcher. To change this icon for your web clipping application, click the **Small** button in the Icons section of the Build PQA dialog box, and then navigate to the image file that you want to use as the small icon.
- A larger icon for users who have chosen Icon view in the Applications Launcher. To change this icon for your web clipping application, click the **Large** button in the Icons section of the Build PQA dialog box, and then navigate to the image file that you want to use as the large icon.

If you don't customize the icons, WCA Builder uses the default icons that are displayed in the Build PQA dialog box.

The icons that the Applications Launcher displays must conform to the following standards:

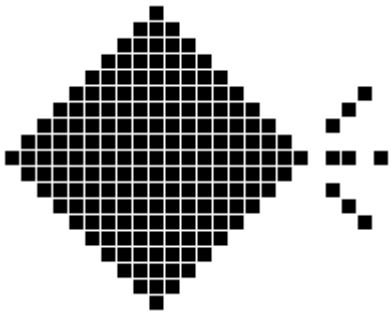
- the maximum size of the small icon is 15x9 pixels
- the maximum size of the large icon is 32x22 pixels
- the image must be in one of the following formats:
  - JPEG
  - GIF
  - BMP

NOTE: Many web clipping applications developers like to include the over-the-air image in their Launcher icons, to help the user identify the application as a web clipping application. If you want to include the diamond image in your icons, you can create the image as follows:

1. Create a 10-pixel by 10-pixel filled square.
2. Rotate the square 45 degrees to create a diamond.
3. Add three “whiskers” to the right corner.

Figure 7.4 shows a magnified version of the over-the-air diamond icon.

Figure 7.4 The Palm over-the-air diamond image



5. Build Your Web Clipping Application

After you have configured the options for your web clipping application, click the Build button in the Build PQA dialog box. WCA Builder compiles all of the local pages and graphics referenced by your top-level page into a `.pqa` file. It stores the generated file in the location that you specified in the dialog box, and optionally copies it into the Install folder for the user you specified.

Using the WCA Builder Command-Line Interface

You can also run the Web Clipping Application Builder from the Windows command line. If you are using the Macintosh version of the program, you cannot use a command line, but a number of these options are available in the graphical user interface of the Macintosh version.

To use the command line interface in Windows, choose Start->Run and type in your command line. If the `wcabuild` program is not in a directory that is part of the current path, you need to browse to its location, or specify its path on the command line.

The syntax for the command line is as follows:

```
wcabuild html-filename [commands] [options]
```

Table 7.3 shows the commands that you can specify on the command line.

Table 7.3 Command line commands for the WCA Builder program

Command	
`/pqa`	Builds the pqa file directly, without user intervention.
`/h`	Displays the WCA Builder help.

Table 7.4 shows the options that you can specify on the command line. Note that Palm provides HTML meta tag equivalents for most of these options. The meta tags are described in Chapter 6, "HTML for Web Clipping Applications."

NOTE: you must use double quotes around options values on the command line.

Table 7.4 Command line options for the WCA Builder program

Option	Meta Tag	Description

`/d <value>`	`PalmPQABitDepth`	Specifies the bit depth to use for images in the built web clipping application.
		
		You can specify one of the following values:
		`"1"`
		`"2"`
		`"4"`
		`"8"`
		`"16"`
		
		The default value is `"2"`.

`/e <value>`	`http-equiv="Content-Type" content="text/html; charset="<value>">`	Specifies the HTML encoding format to use for your web clipping application.
		
		You can specify one of the following values, with case and punctuation ignored:
		`"cp1252"`
		`"shift_jis"`
		
		The default value is `"cp1252"`.

`/l "`*largeicon*`.gif"`	`PalmLargeIconFilename`	Specifies the graphic to use as large icon; use BMP, GIF, or JPEG file. The default value is the large web clipping icon that WCA displays in the Build PQA dialog box.

Table 7.4 Command line options for the WCA Builder program *(continued)*

Option	Meta Tag	Description
`/n "`*appname*`"`	`PalmLauncherName`	Specifies the application name to be displayed in Applications Launcher; this makes the application name different from the `.pqa` file name. WCA Builder may truncate names longer than eight characters. Defaults to the `.pqa` file name.
`/o "`*output*`.pqa"`		Specifies the output file name for your web clipping application.

`/p` <value>	`PalmPostEncoding`	Specifies the POST encoding format to use for your application.
		
		You can specify one of the following values, with case and punctuation ignored:
		`"us-ascii"`
		`"iso-8859-1"`
		`"cp1252"`
		`"shift_jis"`
		`"EUC-JP"`
		`"iso-2022-jp"`
		
		The default value is `"1"`.

`/q` <value>	`PalmPQAVersion`	Specifies which version of the Viewer your application requires. You can specify one of the following values:
		`"1"`
		`"2"`

Table 7.4 Command line options for the WCA Builder program *(continued)*

Option	Meta Tag	Description
`/r <value>`	`PalmLauncherRevision`	Specifies the version of your application, which is displayed in the Applications Launcher **Info** display.
`/s "`*smallicon*`.gif"`	`PalmSmallIconFilename`	Specifies the graphic to use as small icon; use BMP, GIF, or JPEG file. The default value is the small web clipping icon that WCA displays in the Build PQA dialog box.
`/u "`*username*`"`		Specifies user folder into which to install the application
`/v`		WCA Builder runs in verbose mode, which means that it displays error strings.

Using the Image Checker Program

You use the Image Checker program to preview and manipulate images that you want to include in you `.pqa` file. You can resize images and modify their bit depth to optimize their appearance on Palm Powered handheld screens.

NOTE: The editing capabilities of the Image Checker program are restricted to modifying the size and bit-depth of images. You must use an image editing program to make other changes in your images.

The Image Checker is bundled into the MacOS version of the Web Clipping Application Builder. It is a separate application on other operating systems.

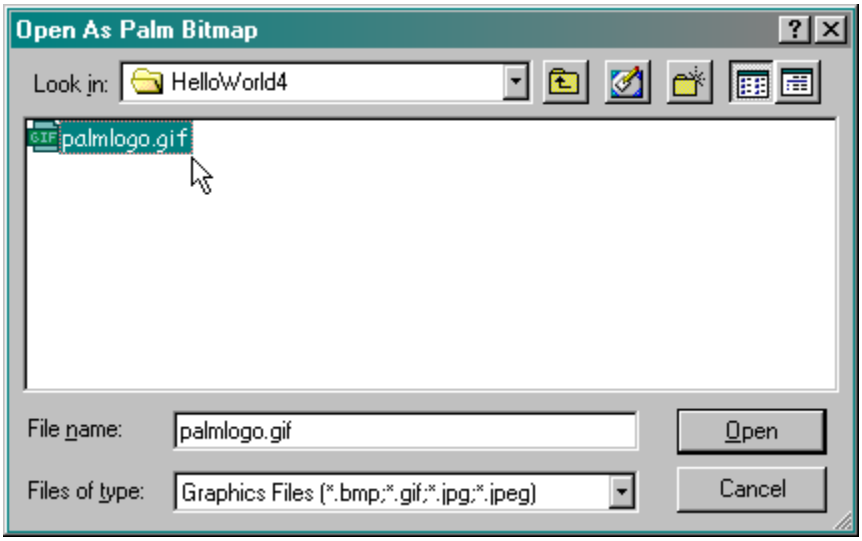
The remainder of this section describes how to use the major operations of the Image Checker:

- Previewing an Image
- Viewing Image Parameters
- Resizing Or Changing the Depth of An Image

Previewing an Image

To preview an image in the Image Checker, you need to select the **Open** command from the File menu. Then navigate to and open the image file, as shown in Figure 7.5.

Figure 7.5 Opening an image in the Image Checker



The Image Checker displays a preview of the image, showing how it will appear on a Palm Powered handheld screen. Figure 7.6 shows an example.

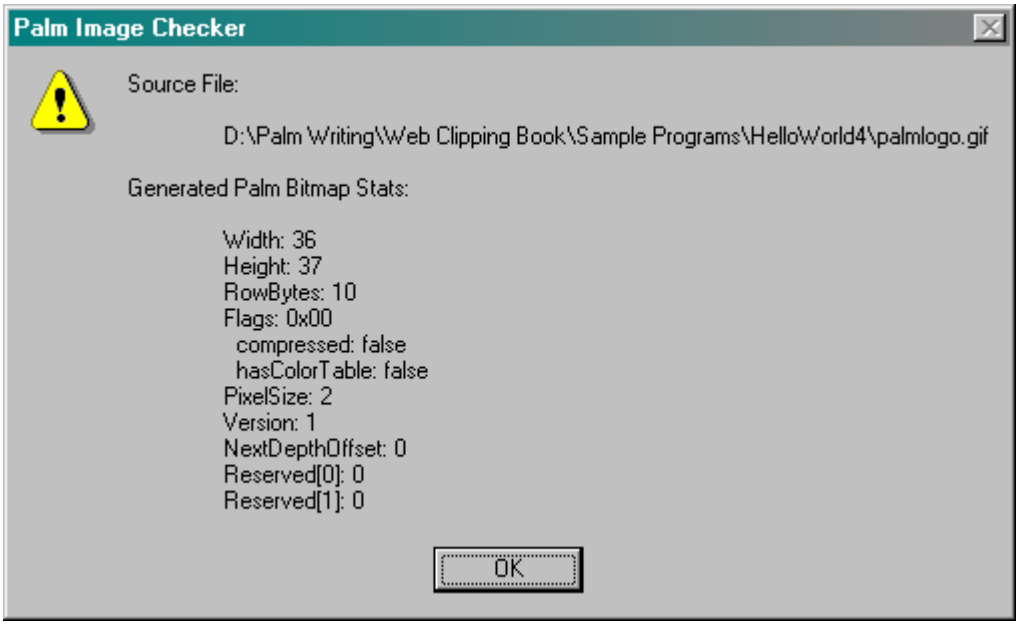
Figure 7.6 The preview of an image

The Image Checker previews the image using the default bit depth, which is 2 bits grayscale display.

Viewing Image Parameters

You can view the parameters of an image that you have opened in the Image Checker by selecting the **View Image Header** command from the Image menu. The Image Checker displays the header information for the image file, as shown in Figure 7.7.

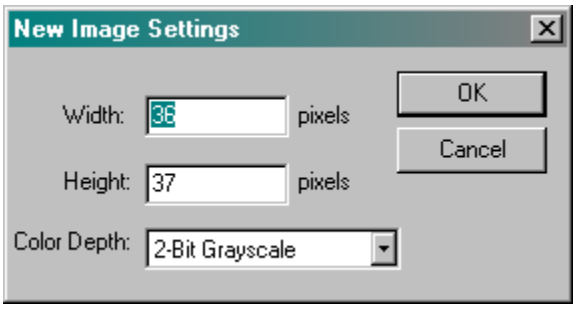
Figure 7.7 Image header display



Resizing Or Changing the Depth of An Image

After you open and preview an image, you can use the Image Checker to resize it or change its color depth. Select the **Resize/ Redepth Image** command from the Image menu, and enter your new width, height, and/or depth values. Figure 7.8 shows the Resize/Redepth dialog box.

Figure 7.8 Resizing an image in the Image Checker



As you can see, the width and height measurements are pixel values. Remember that the content area of Palm Powered Handheld screens is 153 pixels wide by 144 pixels high.

You can select a different maximum bit depth for the image from the pull Color Depth pulldown list.

After you press the OK button, the Image Checker displays a preview of the resized image. You can save this image and, if desired, use a graphics editing program to make other changes to it.

A

Web Clipping Transaction Errors

This appendix provides explanatory information for the common error codes that you may see displayed by the Palm OS® Emulator (POSE) program while testing your web clipping applications.

The following sections describe the errors:

- Proxy Server Error Codes describes the error codes generated by the Palm proxy servers.
- Device Error Codes describes the error codes generated by the Palm OS.
- HTTP Error Codes describes error codes that are generated by the HTTP or HTTPS protocols.
- Miscellaneous Errors describes error codes that do not fit in the other categories.

Proxy Server Error Codes

Table A.1 summarize the error codes generated by the Palm proxy servers; each error is described in the remainder of this section.

Table A.1 Proxy server error codes

Error code	Message
`00002AF9`	Host not found or unavailable
`0000274C`	Timed out
`0000274D`	Connection refused
`C100000C`	The remote machine closed the connection
`C2010005`	An invalid CTP request was received

Table A.1 Proxy server error codes *(continued)*

Error code	Message
C201000D	Attempt to reference an indirect hyperlink with out of date base document
C201001D	Content conversion error - unsupported content type
C201001E	The requested content contained an unsupported encoding type
C2020004	The HTTP response is not formatted correctly
C202000A	Received a request with an unsupported encoding type specified
C202000B	There are no active HTTP or HTTPS proxies available to handle the request
C2060001	Could not correctly recognize internet date string
C2060002	Could not correctly parse an Internet string

Proxy Server Error Code 0002AF9

The message for this error code is "Host not found or unavailable."

This error indicates that one of the following conditions has occurred:

- The proxy server could not find an associated host for the URL given in the activated link or in a redirection link. This can be caused by an improperly formatted HTTP Content-Location header value string.
- The host associated with the given URL is busy.

More recent versions of the clipping proxy server return the HTTP error code 503 instead of this error. For more information, see HTTP Error Code 503.

Proxy Server Error Code 0000274C

The message for this error code is "Timed out."

This error indicates that a response was not received from the Internet server defined by the request URL before the connection time-out expired. Table A.2 shows the current time-out values for various situation.

Table A.2 Server timeout values

Situation	Timeout value
Establishing a connection with an Internet server	30 second
Proxy server receiving a request packet or the complete set of request packets from a Palm Powered handheld	30 seconds
Proxy server receiving a response from a destination server	2 minutes

Proxy Server Error Code 0000274D

The message for this error code is "Connection refused."

This indicates that the proxy server was refused a connection with the Internet server defined in the requested URL. This can occur when the destination server is located behind a firewall that either:

- prevents access through the port given in the URL
- prevents access by requests that were not initiated from a specific set of IP addresses.

If either of these is true, you need to configure the firewall to either:

- allow the clipping proxy server access to the destination server
- allow direct access to the destination server from the Internet

More recent versions of the clipping proxy server return the HTTP error code 503 instead of this error. For more information, see HTTP Error Code 503.

Proxy Server Error Code C100000C

The message for this error code is "The remote machine closed the connection."

This error indicates that the Internet server processing the request closed the connection with the proxy server.

Proxy Server Error Code C2010005

The message for this error code is "An invalid CTP request was received."

This error indicates a problem with the request received by the proxy server or the response received by the device. One possible cause for this error is the use a server-side form (an HTML form sent within a clipping) that includes `SELECT` menus that utilize the `VALUE` attribute in `OPTION` tags.

For example, the following HTML code in a web clipping produces this error when the user attempts to request one of the choices provided in the form>

```
<form method="post" action="http://www.a-make-
believe-domain-name.com/cgi/proc.cgi">
  <select name="example">
    <option value="1">The first choice
    <option value="2">The second choice
    <option value="3">The third choice
    <option value="4">The fourth choice
  </select>
  <input type="submit" value="Go">
</form>
```

To not generate this error, the web clipping needs to use code similar to the following.

```
<form method="post" action="http://www.a-make-
believe-domain-name.com/cgi/proc.cgi">
  <select name="example">
    <option>The first choice
    <option>The second choice
    <option>The third choice
    <option>The fourth choice
  </select>
  <input type="submit" value="Go">
</form>
```

Note that the problem with the use of the `VALUE` attribute in an `OPTION` tag only occurs with server-side forms, not with forms built into a web clipping application.

The cause for this problem relates to the scheme used by the Palm proxy servers to conserve the number of bytes that are transmitted to the device. The proxy servers use hash coding of form fields to avoid sending some of the attributes to the device. What can happen is that the device sends the contents of the VALUE attribute to the proxy server instead of the proper hash code. This makes it impossible for the proxy server to correctly resolve the requested field.

NOTE: This problem has been noted as a bug and will be resolved in the near future.

Proxy Server Error Code C201000D

The message for this error code is "Attempt to reference an indirect hyperlink with out of date base document."

This error indicates that the proxy server attempted could not find a match for the specified URL. This problem can be the result of a problem with the proxy server hash coding scheme.

The Palm proxy servers use a hash code to represent the links as a means of reducing the amount of data that needs to be transmitted to and from the device. This means that when you activate a link, the proxy server converts the hash code back into an URL. To do so, the proxy server re-requests the original clipping that included the link, and uses the hash code to locate the URL within the clipping. The problem occurs when the location of the link in the clipping has changed since the hashing was performed.

NOTE: Starting with version 4.0 of the Palm OS, you can specify that you don't want specific (or all) links hash-coded in your pages with The PalmHREFStyle Meta Tag.

Proxy Server Error Code C201001D

The message for this error code is "Content conversion error - unsupported content type."

This error usually indicates that the page includes a media type that is not supported by the Palm proxy servers. Currently, only the following MIME types are supported:

- text/html
- text/plain
- image/gif
- image/jpeg

For example, this error is generated if a server sends an HTML message containing the header "Content-Type: text/html;charset=ISO-8859-1."

Proxy Server Error Code C201001E

The message for this error code is "The requested content contained an unsupported encoding type."

This error results from the HTTP header "Content-encoding" being included in the response. Currently, no content encoding is supported for responses (clippings).

Proxy Server Error Code C2020004

The message for this error code is "The HTTP response is not formatted correctly."

This error can result from the proxy server receiving an HTML or HTTPS response message that it finds invalid, possibly because of an invalid HTTP(S) header field. HTTP(S) headers should conform to the RFC 2616 standard, *Hypertext Transfer Protocol -- HTTP/1.1*.

For example, the Date header fields should conform to the specification as outlined in sections 4.2 and 14.8 of RFC 2616.

Proxy Server Error Code C202000A

The message for this error code is "Received a request with an unsupported encoding type specified."

This error indicates that an unsupported encoding type was used. Currently, the proxy servers support only the following encoding types:

- `application/x-www-form-urlencoded`

Note that this is the default encoding type for an HTML form.

Proxy Server Error Code C202000B

The message for this error code is "There are no active HTTP or HTTPS proxies available to handle the request."

This error is generated by the Palm proxy server when it is unable to connect to a proxy cache. This usually indicates a problem with a proxy cache at Palm.Net™.

If you receive this error, notify Palm.Net Technical Support of the problem by e-mail at support@palm.net or by telephone at (407) 531-4400. If you received this error while using the emulator, be sure to inform the support people about which proxy server IP address you were using in POSE.

Proxy Server Error Code C2060001

The message for this error code is "Could not correctly recognize internet date string."

This error results when the HTTP(S) Date header contains a string that is not recognized by the clipping proxy server. Your HTTP and HTTPS headers must conform to the RFC 2616 standard, *Hypertext Transfer Protocol -- HTTP/1.1*.

NOTE: The value `0` is not legitimate for a date string, according to the HTTP 1.1 specification.

Proxy Server Error Code C2060002

The message for this error code is "Could not correctly parse an Internet string."

This error indicates that the proxy server has a problem parsing something in the page. The problem could be with:

- a requested URL string
- the format of an HTML header string
- a blank HTTP header value

Your HTTP and HTTPS headers must conform to the RFC 2616 standard, *Hypertext Transfer Protocol -- HTTP/1.1*.

Device Error Codes

Table A.3 summarize the error codes generated by the Palm OS; each error is described in the remainder of this section.

Table A.3 Device error codes

Error code	Message
`Net 1205`	No more sockets
`Net 120C`	No interfaces
`INet 1410`	Your handheld could not connect to the server
`INet 1413`	Your handheld could not connect to the server
`INet 145D`	Your handheld lost connection with the server

NetLib Error Code Net 1205

The message for this error code is "No more sockets."

This error indicates that the maximum number of sockets that can be open at any one time has been reached. There are currently a maximum of 4 TCP/IP sockets allowed open by the Palm OS. To avoid this error, your code must close an open socket when it is no longer needed. It is also a good idea to check for the number of currently open sockets before attempting to open a new socket.

NetLib Error Code Net 120C

The message for this error code is "No interfaces."

This error is usually the result of not having configured the emulator to redirect NetLib requests to the desktop computer's TCP/IP connection, as described in 4. Configure POSE for Web Clippings in Chapter 2, "Getting Started with Web Clippings."

INetLib Error Code INet 1410

The message for this error code is "Your handheld could not connect to the server. Wait a few minutes and try again."

This error indicates a problem connecting to the Palm proxy server. If you are using POSE, try checking the configuration, as described in 4. Configure POSE for Web Clippings in Chapter 2, "Getting Started with Web Clippings."

INetLib Error Code INet 1413

The message for this error code is "Your handheld could not connect to the server. Wait a few minutes and try again."

This error indicates that a connection was established to the proxy server or destination server, but a read timeout occurred before any data was received in the socket buffer.

INetLib Error Code INet 145D

The message for this error code is "Your handheld lost connection with the server. Wait a few minutes and try again."

This error indicates that the connection between the device or emulator and the server (proxy server or destination server) has been closed. The problem could reside with the servers or the network.

One possible cause for this error is that the Palm OS issues a partial shutdown command for the port, which some firewalls consider a security risk; these firewalls then completely shut down the port, so the information gets stalled at the firewall and never reaches the Palm Powered handheld.

HTTP Error Codes

Table A.4 summarizes the HTTP(S) error codes that you might see while debugging your web clipping applications. The descriptions in this chapter have been extracted from the RFC 2616 standard, *Hypertext Transfer Protocol -- HTTP/1.1*. Each error is described in the remainder of this section.

Table A.4 HTTP error codes

Error code	Message
400	Bad Request
401	Unauthorized
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URI Too Long
415	Unsupported Media Type
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable

Table A.4 HTTP error codes *(continued)*

Error code	Message
504	Gateway Timeout
505	HTTP Version Not Supported

HTTP Error Code 400

The message for this error code is "Bad Request."

The request could not be understood by the server due to malformed syntax.

HTTP Error Code 40

The message for this error code is "Unauthorized."

The request requires user authentication.

HTTP Error Code 403

The message for this error code is "Forbidden."

The server understood the request, but is refusing to fulfill it.

HTTP Error Code 404

The message for this error code is "Not Found."

The server has not found anything matching the Request-URI.

HTTP Error Code 405

The message for this error code is "Method Not Allowed."

The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.

HTTP Error Code 406

The message for this error code is "Not Acceptable."

The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

HTTP Error Code 407

The message for this error code is "Proxy Authentication Required."

This code is similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy.

HTTP Error Code 408

The message for this error code is "Request Timeout."

The client did not produce a request within the time that the server was prepared to wait.

HTTP Error Code 409

The message for this error code is "Conflict."

The request could not be completed due to a conflict with the current state of the resource.

HTTP Error Code 410

The message for this error code is "Gone."

The requested resource is no longer available at the server and no forwarding address is known.

HTTP Error Code 411

The message for this error code is "Length Required."

The server refuses to accept the request without a defined Content-Length.

HTTP Error Code 412

The message for this error code is "Precondition Failed."

The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

HTTP Error Code 413

The message for this error code is "Request Entity Too Large."

The server is refusing to process a request because the request entity is larger than the server is willing or able to process.

HTTP Error Code 414

The message for this error code is "Request-URI Too Long."

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

HTTP Error Code 415

The message for this error code is "Unsupported Media Type."

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

HTTP Error Code 417

The message for this error code is "Expectation Failed."

The expectation given in an Expect request-header field (see section 14.20) could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

HTTP Error Code 500

The message for this error code is "Internal Server Error."

The server encountered an unexpected condition which prevented it from fulfilling the request.

HTTP Error Code 501

The message for this error code is "Not Implemented."

The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.

HTTP Error Code 502

The message for this error code is "Bad Gateway."

The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

HTTP Error Code 503

The message for this error code is "Service Unavailable."

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

For more information, see Proxy Server Error Code 0002AF9 or Proxy Server Error Code 0000274D.

HTTP Error Code 504

The message for this error code is "Gateway Timeout."

The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request.

HTTP Error Code 505

The message for this error code is "HTTP Version Not Supported."

The server does not support, or refuses to support, the HTTP protocol version that was used in the request message. The server is indicating that it is unable or unwilling to complete the request using the same major version as the client, as described in section 3.1, other than with this error message.

Miscellaneous Errors

This section describes the errors that do not fit into the other categories in this appendix.

Web Clipping Application Viewer Errors

The Web Clipping Application Viewer (Viewer) sometimes generates errors when running in POSE. In almost all cases, these errors do not occur when using a physical device, and you can simply click **Continue** to continue testing your web clipping application.

The following is an example of a Viewer error message as displayed in POSE:

Web Clipping 3.2 has just read directly from memory manager data structures.

B

Palm HTML Summary

This appendix provides tables of all HTML markup compliant with the Palm OS® Web Clipping Application Viewer (the Viewer), including Palm extensions. Many of the HTML elements described here are discussed in Chapter 6, "HTML for Web Clipping Applications."

For the complete Document Type Description (DTD) of the HTML subset used with the Viewer, see Appendix C, "Palm OS HTML 3.2 DTD."

HTML Structural Tags

Table B.1 shows the structural tags that you can use in your web clipping HTML pages.

Table B.1 Structural tags

Tag	Description	Attributes	Notes
<BODY>	Document content	BGCOLOR, TEXT	**Palm recommends in all documents.**
<HEAD>	Encloses the document header		**Palm recommends in all documents**
<HTML>	Indicates beginning of HTML source		**Palm recommends in all documents**
<META>	Used to include document-level information	HTTP-EQUIV, NAME, CONTENT	See Table B.2 for Palm-specific META tags.
<TITLE>	Text appears in Navigation bar		**Palm recommends in all documents**

HTML Palm-Specific Elements

Table B.2 shows the Palm-specific tags, variables, and input types that you can use in your HTML pages. Each of these entities is described in detail in Chapter 6, "HTML for Web Clipping Applications."

Table B.2 Palm-specific elements

Tag	Description	Notes
`%DeviceId`	Variable replaced by unique device ID of Palm Powered handheld for use in CGI scripts.	See The %DeviceID Variable in Chapter 6.
`%Location`	Variable replaced by information about the location of the connected handheld device.	See The %Location Variable in Chapter 6.
`%WCDevCaps`	Variable replaced by information about the capabilities of the connected handheld device.	See The %WCDevCaps Variable in Chapter 6.
`%ZipCode`	Variable replaced by the postal code of the local base station	See The %Zipcode Variable in Chapter 6.
`<SmallScreenIgnore>`	Tag that marks areas that are not displayed by the Viewer	**Use in clippings.** See The SmallScreenIgnore Tag in Chapter 6.
`Button`	Anchor attribute that produces a a labeled hyperlink button	See The Button Tag in Chapter 6.

Table B.2 Palm-specific elements *(continued)*

Tag	Description	Notes
`DatePicker`	Form input type that displays the Palm date selector.	See The DatePicker Input Type in Chapter 6.
`HistoryListText`	META tag that customizes History list	**Palm recommends for all clippings.** See The HistoryListText Meta Tag in Chapter 6.
`LocalIcon`	META tag that includes local graphics and HTML pages in application	**Use only in application pages.** See The LocalIcon Meta Tag in Chapter 6.
`Palm`	Tag to launch a Palm OS application on the handheld device.	See The Palm and PalmCall Tags in Chapter 6.
`PalmBinary`	Attribute used in URLs to specify a file for download to a Palm Powered handheld.	See The PalmBinary Attribute in Chapter 6.
`PalmCall`	Tag to sublaunch a Palm OS application on the handheld device.	See The Palm and PalmCall Tags in Chapter 6.
`PalmComputingPlatform`	META tag that identifies the page as Palm-friendly	**Required in all documents.** Use to ensure that the Viewer displays your text and images properly. See The PalmComputingPlatform Meta Tag in Chapter 6.

Table B.2 Palm-specific elements *(continued)*

Tag	Description	Notes
`PalmDoNotCache`	META tag to prevent caching of sensitive data	See The PalmDoNotCache Meta Tag in Chapter 6.
`PalmHREFStyle`	META tag to prevent link hashing on a web clipping page.	See The PalmHREFStyle Meta Tag in Chapter 6.
`PalmLargeIconFilename`	META tag that specifies the name of the icon file to use in the icon view in the Palm OS Applications Launcher.	See The PalmLargeIconFilename and PalmSmallIconFilename Meta Tags in Chapter 6.
`PalmLauncherName`	META tag that specifies the name to display for your web clipping application in the Palm Applications Launcher.	See The PalmLauncherName Meta Tag in Chapter 6.
`PalmLauncherRevision`	META tag that includes version number in application	**Palm recommends for the index page of web clipping applications.** See The PalmLauncherRevision Meta Tag in Chapter 6.
`PalmPostEncoding`	META tag that specifies the encoding method to use for POSTed data.	See The PalmPostEncoding Meta Tag in Chapter 6.
`PalmPQABitDepth`	META tag that specifies the maximum bit depth of images in your web clipping.	See The PalmPQABitDepth Meta Tag in Chapter 6.

Table B.2 Palm-specific elements *(continued)*

Tag	Description	Notes
`PalmSmallIconFilename`	META tag that specifies the name of the icon file to use in the list view in the Palm OS Applications Launcher.	See The PalmLargeIconFilename and PalmSmallIconFilename Meta Tags in Chapter 6.
`PalmStyle`	Attribute used in URLs to specify that the proxy server should use full or indexed (hashed) URL specifications.	See The PalmStyle Attribute in Chapter 6.
`TimePicker`	Form input type that displays the Palm time selector.	See The TimePicker Input Type in Chapter 6.

HTML Block Tags

Table B.3 shows the block tags you can use in your web clipping HTML pages.

Table B.3 Block tags

Tag	Description	Attributes	Notes
`<ADDRESS>`	Encloses author and contact details		
`<BLOCKQUOTE>`	Indents block quotations		
` `	Forces line break	`CLEAR`	`CLEAR` moves break down past floating images on either or both margins.
`<CENTER>`	Same as `<DIV ALIGN=CENTER>`		

Table B.3 Block tags *(continued)*

Tag	Description	Attributes	Notes
<DIV>	Structures document with hierarchical divisions; breaks up <P> with a new line	ALIGN	
<H1>-<H6>	Hierarchical documents headings	ALIGN	
<HR>	Horizontal rule	ALIGN, SIZE, WIDTH	SIZE changes the thickness of the line. WIDTH defaults to 50%. Values can be expressed as a percentage or as the number of pixels.
<LISTING>	Includes preformatted ASCII text	None	Deprecated but supported. Replace with PRE.
<P>	Paragraph break	ALIGN	
<PLAINTEXT>	Includes preformatted ASCII text	None	Deprecated but supported. Replace with PRE.
<PRE>	Includes preformatted ASCII text.	None	
<XMP>	Includes preformatted ASCII text	None	Deprecated but supported. Replace with PRE.

HTML Formatting Tags

Table B.4 shows the formatting tags you can use in your web clipping HTML pages.

Table B.4 Formatting tags

Tag	Description	Notes
<B>	Boldface	See Fonts on Web Clipping Pages in Chapter 6.
<BIG>	Increase size	
<CITE>	Italic	
<CODE>	Monospace	
<DFN>	Italic	
<EM>	Italic	
<FONT>	Varies	Can use the SIZE and COLOR attributes. See Fonts on Web Clipping Pages in Chapter 6.
<I>	Italic	
<KBD>	Monospace	
<S>	Strikethrough	
<SAMP>	Monospace	
<SMALL>	Decrease size	
<STRIKE>	Strikethrough	
<STRONG>	Boldface	
<TT>	Monospace	
<U>	Underline	
<VAR>	Italic	

HTML Hypertext and Image Tags

Table B.5 shows the hypertext and image tags you can use in your web clipping HTML pages.

Table B.5 Hypertext and image tags

Tag	Description	Attributes	Notes
<A>	Hyperlink or anchor	NAME, HREF, TITLE, BUTTON	See Palm Tags for Use in URL Specifications in Chapter 6.
<BASE>	BASE URL dereferences relative URLs	HREF=<*URL*>	Use to dereference relative URLs.
<IMG>	Inserts image	SRC, ALT, HEIGHT, WIDTH, ALIGN	See Images on Web Clipping Pages in Chapter 6.

HTML List Tags

Table B.6 shows the list tags you can use in your web clipping HTML pages.

Table B.6 List tags

Tag	Description	Attributes	Notes
<DD>	Definition description		Use with <DL> and <DT>
<DL>	Definition list		Use with <DT> and <DD>
<DT>	Definition title		Use with <DL> and <DD>
<LI>	List item	TYPE, VALUE	Use with <UL> or <OL>

Table B.6 List tags *(continued)*

Tag	Description	Attributes	Notes
<OL>	Ordered list	TYPE, START	Viewer displays a maximum of five levels of list. Nested lists are supported, up to the width of the screen.
<UL>	Unordered list	TYPE, START	Viewer displays a maximum of five levels of list. Nested lists are supported, up to the width of the screen.

HTML Table Tags

Table B.7 shows the table tags you can use in your web clipping HTML pages.

Table B.7 Table tags

Tag	Description	Attributes	Notes
<CAPTION>	Table caption	ALIGN	
<TABLE>	Starts and ends table	ALIGN, WIDTH, BORDER, CELLPADDING, CELLSPACING	Tables are rendered without a border, unless BORDER is specified. Tables are sized to "best-fit" unless WIDTH is set in pixels. Nested tables are not supported.

Table B.7 Table tags *(continued)*

Tag	Description	Attributes	Notes
<TD>	Table cell (data)	ALIGN, ROWSPAN, WIDTH, COLSPAN, HEIGHT	
<TH>	Table cell (header)	ALIGN, ROWSPAN, WIDTH, COLSPAN, HEIGHT	
<TR>	Table row	ALIGN	

HTML Form Tags

Table B.8 shows the form tags you can use in your web clipping HTML pages.

Table B.8 Form tags

Tag	Description	Attributes	Notes
<FORM>	Begins and ends form	METHOD, ACTION, ENCTYPE	The POST and GET methods are supported as is the default ENCTYPE of application/x-wwww-form-url-encoded.
<INPUT>	Specifies input fields in form	TYPE, NAME, VALUE, CHECKED, SIZE, MAXLENGTH, SRC, ALIGN	For more information, see Using Forms in Web Clipping Applications in Chapter 4.
<OPTION>	Item in pop-up menu	SELECTED, VALUE	

Table B.8 Form tags *(continued)*

Tag	Description	Attributes	Notes
<SELECT>	Multiple-choice selections in pop-up menu	NAME, SIZE, MULTIPLE	
<TEXTAREA>	Multiline input field	ROWS, COLS, NAME	

HTML Form Input Types and Attributes

Table B.9 shows the input types and attributes you can use in forms in your web clipping HTML pages.

Table B.9 Form input types

Input type	Description	Attributes
CHECKBOX	Check boxes for multiple selections	NAME, VALUE, CHECKED
DATEPICKER	Displays the Palm OS date selector	
HIDDEN	Unrendered field whose information is to be sent back to the server on submission of form	NAME, VALUE
PASSWORD	Single-line text field; provides a box that displays either "Unassigned" to prompt for password or "Assigned" to show password entered	NAME, VALUE, SIZE, MAXLENGTH
RADIO	Radio buttons for exclusive selection	NAME, VALUE, CHECKED
RESET	Resets form to initial state	NAME, VALUE
SUBMIT	Submit button to send form contents to server	NAME, VALUE
TEXT	Single-line text field	NAME, VALUE, SIZE, MAXLENGTH
TIMEPICKER	Displays the Palm OS time selector	

C

Palm OS HTML 3.2 DTD

This appendix provides the complete Palm OS® HTML 3.2 Document Type Description (DTD). For more information about the specifics of HTML for web clippings, see Chapter 6, "HTML for Web Clipping Applications."

Listing C.1 Palm OS HTML 3.2 DTD

```
Palm OS HTML 3.2 Document Type Definition

<!--
Palm OS Document Type Definition for the HyperText Markup Language version 3.2
as implemented for the Web Clipping Applicaton Viewer of Palm OS 4.0 and the
Web Clipping proxy server.

For more information on the Palm OS look at URL http://www.palmos.com

Date: Friday, February 9, 2001

Authors:  Kathy Ly <kathy.ly@palm.com>, and Scot Stennis based on the W3C HTML
3.2 DTD by Dave Raggett  <dsr@w3.org>

Palm OS HTML 3.2 aims to capture suppored W3C HTML 3.2 elements and attributes
as swell as Palm OS extensions.
-->

<!ENTITY % HTML Version "-//POS//DTD Palm OS HTML 3.2 Final//EN"

--Typical usage (optional):

<!DOCTYPE HTML PUBLIC "-//POS//DTD Palm OS HTML 3.2 Final//EN">
<html>
...
<html>

-->
```

```
<!--=======================Deprecated Features Switch=======================-->

<!ENTITY % HTML.Deprecated "INCLUDE">


<!--============================Imported Names================================-->

<!ENTITY % Content-Type "CDATA"
--meaning a MIME content type, as per RFC1521
-->

<!ENTITY % HTTP-Method "GET | POST"
--as per HTTP specification
-->

<!ENTITY % URL "CDATA"
--The term URL means a CDATA attribute whose value is a Uniform Resource
   Locator, See RFC1808 (June 95) and RFC1738 (Dec 94).
-->

<!-- Parameter Entities -->

<!ENTITY % head.misc "META -- repeatable head elements -->

<!ENTITY % heading "H1|H2|H3|H4|H5|H6">

<!-- DIR and MENU behave as UL -->
<!ENTITY % list "UL | OL | DIR |MENU">

<![ %HTML.Deprecated [
  <!ENTITY % preformatted "PRE | XMP | LISTING">
]]>

<!ENTITY % preformatted "PRE">

<!--====================== Character mnemonic entities =====================-->

<!ENTITY % ISOlat1 PUBLIC
  "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
%IOSlat1;


<!--====================== Entities for special symbols ====================-->

<!-- Some common symbols ( e.g.: &reg; ): -->

<!ENTITY ampCDATA "&#38;"-- ampersand-->
<!ENTITY gt CDATA "&#62;"-- greater than-->
```

```
<!ENTITY lt
<!ENTITY copy
<!ENTITY reg


<!--====================== Text Markup ====================================-->

<!-- S and STRIKE are synonymous -->
<!ENTITY % font "TT | I | B | U | STRIKE | S | BIG | SMALL">

<!-- EM, DFN, VAR, CITE have similar behavior;
     CODE, SAMP, KBD have similar behavior;
     STRONG behaves as B
-->
<!ENTITY % phrases "EM | STRONG | DFN | CODE | SAMP | KBD | VAR | CITE">

<!ENTITY % special "A | IMG | FONT | BR">

<!ENTITY % form "INPUT | SELECT | TEXTAREA">

<!ENTITY % text "#PCDATA | %font | %phrase | %special | %form">

<!ELEMENT (%font | %phrase) - - (%text)*>

<!--The "web safe" palette of colors are now supported
-->

<!ELEMENT FONT - - (%text)*  --local change to font -->
<!ATTLIST FONT
  size     CDATA           #IMPLIED  --[+]nn e.g. size="+1", size=4--
  color    CDATA           #IMPLIED  --#RRGGBBS in hex, e.g. black=#000000#
-->

<!ELEMENT BASEFONT  - O  EMPTY  -- base font size (1 to 6) -->
<!ATTLIST BASEFONT
  size     CDATA           #IMPLIED  -- e.g.: size=3 -->

<!ELEMENT BR - O EMPTY -- forced line break -->
<!ATTLIST BR
  clear (left | all | right | none) none -- control of text flow -->

<!--======================== HTML content models ============================-->
<!--
  HTML has three basic content models:
     %text              character level elements and text strings
     %flow              block-like elmenets e.g. paragraphs and lists
     %body.content      as %flow plus headers H1-H6 and ADDRESS
```

```
-->

<!ENTITY % block
  "P | %list | %preformatted | DL | DIV | CENTER | BLOCKQUOTE | FORM | HR |
TABLE">

<!-- %flow is used for DD and LI -->
<!ENTITY % flow "(%text | %block)*">

<!--========================= Document Body ================================-->

<!ENTITY % body.content "(&heading | %text | %block | ADDRESS)*">

<!ENTITY % color "CDATA" -- a color specification: #HHHHHH -->

<!ENTITY % body-color-attrs "
   bgcolor %color#IMPLIED
   text%color#IMPLIED
   ">

<!ELEMENT BODY O O %body.content>
<!ATTLIST BODY
   %body-color-attrs;  -- bgcolor, text --
   >

<!ENTITY % ADDRESS.content "((%tesxt;) | P)*">

<!ELEMENT ADDRESS - - %address.content>

<!ELEMENT DIV -- %body.content>
<!ATTLIST DIV
   align(left | center | right)#IMPLIED -- alignment of following text --
   >

<!-- CENTER is a shorthand for DIV with ALIGN= CNETER -->
<!ELEMENT center - - %body.content>

<!--========================== The Anchor Element ==========================-->

<!ELEMENT A - - (%text)* -(A)>
<!ATTLIST A
   name       CDATA           #IMPLIED  -- named link end --
   href       %URL            #IMPLIED  -- URL for linked resource --
   title      CDATA           #IMPLIED  -- advisory title string --
   button     (button)        #IMPLIED  -- used to created the button image for
                                          the link, Palm-specific --
```

```
   PalmStyle  (index|full)    #IMPLIED  -- Palm-specific attribute to determine
                                        how to send a link to a device, "index"
                                        is the default value --
   PalmBinary (PalmBinary)    #IMPLIED  -- Palm-specific attribute to specify a
                                           binary file for download to device -
->

<!--============================ Images ===================================-->

<!ENTITY % Length "CDATA"-- nn for pixels or nn% for percentage length -->
<!ENTITY % Pixels "NUMBER"-- integer representing length in pixels -->

<!-- Suggested widths are used for negotiating image size with the module
responsible for painting the images.  align=left or right cause image to float
to margin anf for subsequent text to wrap around image -->

<!ENTITY % IAlign "(top | middle | bottom | left | right)">

<!ELEMENT IMG- O    EMPTY-- Embedded image -->
<!ATTLIST IMG
   src     %URL       #REQUIRED -- URL of image to embed --
   alt     CDATA      #IMPLIED  -- for display in place of image --
   align   %IAlign    #IMPLIED  -- vertical or horizontal alignment --
   height  %Pixels    #IMPLIED  -- suggested height in pixels --
   width   %PIxels    #IMPLIED  -- suggested width in pixels --
   >

<!--========================== Horizontal Rule ============================-->

<!ELEMENT HR- o     EMPTY>
<!ATTLIST HR
   align   (left | right | center)    #IMPLIED
   size    %Pixels                    #IMPLIED
   width   %Length                    #IMPLIED
   >

<!--============================ Paragraphs ===============================-->

<!ELEMENT P- O(%text)*>
<!ATTLIST P
   align   (left | center | right)    #IMPLIED
   >

<!--============================ Headings =================================-->

<!--
There are six levels of headers from H1 (the most important) to H6 (the least
important).
```

```
-->

<!ELEMENT ( %heading ) - -  (%text;)*>
<!ATTLIST ( %heading )
   align (left | center | right)    #IMPLIED
   >

<!--============================== Preformatted Text ========================-->

<!-- excludes images and changes in font size -->
<!ENTITY % pre.exclusion "IMG| BIG | SMALL | FONT">

<!ELEMENT PRE - - (%text)* -(%pre.exlusion)>

<![ %HTML.Deprecated [

<!ENTITY % literal "CDATA"
   -- historical, non-conforming parsing mode where the only
           markup signal is the end tage in full
        -->

<!ELEMENT (XMP | LISTING) - - %literal>
<!ELEMENT PLAINTEXT - o %literal>

]]>

<!--============================== Block-like Quotes ========================-->

<!ELEMENT BLOCKQUOTE - - % body.content>

<!--============================== Lists ====================================-->

<!--

Palm OS HTML 3.2 allows you to control the sequence number for ordered lists.
You can set the sequence number width the START and VALUE attributes.  The TYPE
attribute may be used to specify the rendering of ordered and unordered lists.

-->

<!-- definition lists - DT for term, DD for its definition -->
<!ELEMENT DL - - (DT | DD)+>
<!ATTLIST DL>

<!ELEMENT DT - O (%text)*>
<!ELEMENT dd - O %flow;>

<!-- Ordered lists OL, and unordered lists UL -->
```

```
<!ELEMENT (OL | UL) - - (LI)+>

<!--
   Numbering style
   1  Arabic numbers    1, 2, 3, ...
   a  lower alpha       a, b, c, ...
   A  upper alpha       A, B, C, ...
   i  lower Roman       i, ii, iii, ...
   I  upper Roman       I, II, III, ...

   The style is applied to the sequence number which by default is reset
   to 1 for th efirst list item in an ordered list.

-->

<!ENTITY % OLStyle "CDATA" -- contrainsed to: [1 | a | A | i | I] -->

<!ATTLIST OL -- ordered lists --
   type    %OLStyle           #IMPLIED -- NUMBERING STYLE --
   START   NUMBER             #IMPLIED -- starting sequence number --
   >

<!-- bullet styles -->

<!-- disc and square are rendered as square; disc is the default type-->
<!ENTITY % ULStyle "disc | square | circle">

<!ATTLIST UL -- unordered lists --
   type    %ULStyle           #IMPLIED -- bullet style --
   >

<!ELEMENT (DIR | MENU) - - (LI)+ -(%block)>
<!ATTLIST DIR
   >
<!ATTLIST MENU
   >

<!-- <DIR>    Directory list   -->
<!-- <MENU>   Menu list   -->

<!-- The type attribute can be used to change the bullet style in unordered
lists and the numbering styple in ordered lists -->

<!ENTITY %LIStyle "CDATA" -- constrained to : "(%ULStyle | % OLStyle)" -->

<!ELEMENT LI - O %flow  -- list item -->
<!ATTLIST LIT
```

```
   type    %LIStyle           #IMPLIED -- list item style --
   value   NUMBER             #IMPLIED -- reset sequence number --
   >

<!--========================== Forms ====================================-->

<!ELEMENT FORM - - %body.content -(FORM)>
<!ATTLIST FORM
   action  %URL            #IMPLIED -- server-side form handler --
   method  (%HTTP_Mehtod)  GET      -- see HTTP specification --
   enctype %Content-Type;  "application/x-www-form-urlencoded"
   >

<!ENTITY % InputType
   "(TEXT | PASSWORD | CHECHBOX | RADIO | SUBMIT | RESET | HIDDEN |
   DATEPICKER | TIMEPICKER)"
              -- DATEPICKER displays MM/DD//YY format,
                 but submits as YYY_MM_DD format.
                 TIMPICKER displays HH:MM (am/pm) format,
                 but submits 24 hour clock HH:MM format.
-->

<!--There are four Palm-specific tags that can be used in conjunction with
hidden fields:
     -%DeviceID, which will return the unique ID associated with the device
     %Location, which will return location information from the carrier
                 network, if available
     -%WCDevCaps, which will return device capability information
     -%Zipcode, which will return the zipcode of the base station with which
                 the user is currently communicating
-->

<!ELEMENT INPUT - O EMPTY>
<!ATTLIST INPUT
   type    %InputType    TEXT       -- what kind of widget is needed --
   name    CDATA         #IMPLIED   -- required for all but submit and reset --
   value   CDATA         #IMPLIED   -- required for radio and checkboxes --
   checked (checked)     #IMPLIED   -- for radio buttons and check boxes --
   size    CDATA         #IMPLIED   -- specific to each type of field --
   maxlength NUMBER      #IMPLIED   -- max chars allowed in text fields --
   >

<!ELEMENT SELECT - - (OPTION+)>
<!ATTLIST SELECT
   name   CDATA          #REQUIRED
   size   NUMBER         #IMPLIED
   multiple (multiple)   #IMPLIED -- size attribute should be set to minimum of
                                  2 for proper display when using multiple --
```

```
   >

<!ELEMENT OPTION - O (#PCDATA)*>
<!ATTLIST OPTION
   selected (selected)   #IMPLIED
   value  CDATA          #IMPLIED   -- defaults to element content --
   >

<!-- Multi-line text input field. -->
<!ELEMENT TEXTAREA - - (#PCDTA)*>
<!ATTLIST TEXTAREA
   name CDATA            #REQUIRED
   rows NUMBER           #REQUIRED
   cols NUMBER           #REQUIRED
   >

<!--========================== Tables ===================================-->

<!-- Widely deployed subset of the full table standard, see RFC 1942
     e.g. at http://www.ics.uci.edu/pub/ietf/html/rfc1942.txt -->

<!-- horizontal placement of table relative to window -->
<! ENTITY % Where "(left | center | right)">

<!-- horizontal alignemnt attributes for cell contents -->
<!ENTITY % cell.halign
   "align (left | center | right)     #IMPLIED
   >

<!ELEMENT table - - (caption?, tr+)>
<!ELEMENT tr - O (th| td)*>
<!ELEMENT (th|td) - O %body.content>

<!ATTLIST table
   align         %Where     #IMPLIED  -- table position relative to window --
   width         %Length    #IMPLIED  -- table width relative to window;
                                           percentage widths NOT support --
   border        %Pixels    #IMPLIED  -- controls frame width around table --
   cellspacing   %Pixels    #IMPLIED  -- spacing between cells --
   cellpadding   %Pixels    #IMPLIED  -- spacing within cells --
   >

<!ELEMENT CAPTION - - (%text;)*   -- table or figure caption -->
<!ATTLIST CAPTION
   align (top | bottom) #IMPLIED
   >

<!ATTLIST tr      -- table row --
```

```
   %cell.halign;   -- horizontal alignment in cells--
   >

<!ATTLIST (th|td)  -- header or data cell
   rowspan         NUMBER  1           -- number of rows spanned by cell --
   colspan         NUMBER  1           -- number of cols spanned by cell --
   %cell.halign;                       -- horizontal alignment in cell --
   width           %Pixels#IMPLIED  -- suggested width for cell --
   height          %Pixels#IMPLIED  -- suggested height for cell --
   >

<!--========================== Palm Special ================================-->

<!ELEMENT SMALLSCREEN IGNORE - - %body.content>

<!--========================== Document Head ===============================-->

<!-- %head.misc defined earlier on as "META" -->

<!-- ENTITY % head.content "TITLE & BASE?">

<!ELEMENT HEAD O O (%head.content ) +(%head.misc)>

<!ELEMENT TITLE - - (#PCDATA)* - (%head.misc)
   -- The TITLE element is not considered part of the flow of text.
      It is diplayed in the menu-bar as the screen title.
   -->

<!--
    The BASE element gives an absolute URL for dereferencing relative URLs,
e.g.

   <BASE href = "http://foo.com/">
   ...
   <IMG SRC= "images/bar.gif">

    The image is deferenced to http://foo.com/images/bar.gif

In the absence of a BASE element the document URL should be used.  Note that
this is not necessarily the same as the URL used to request the document, as
the base URL may be overridden by an HTTP header accompanying the document.
-->

<!ELEMENT BASE - O EMPTY>
<!ATTLIST BASE
   href %URL  #REQUIRED
   >
```

```
<!--
Palm OS special name/content pairs:
   name = "PalmComputingPlatform" content="true"
              -identifies as Palm Friendly and geared for the device's small
               screen size
   name = "HistoryListText" content="some string"
              -used by clippings to show significant info in History List
   name = "LocalIcon" content="filename"
              -used to include images or html pagse into the web clipping
              application during a build when those images or pages aren't
              referenced by any of the local pages, thus allowing clipping to
use
              them
   name = "PalmDoNotCache" content="true"
              -prevents caching of sensitive data
   name = "PalmHREFStyle" content="value"
              -value is either FULL or INDEXED, used to send either indexed
              (default) or full URL links from the Palm proxy servers to the
              device

Palm OS pairs that are only relevant for use in pages that will be built by the
WCA Builder:

   name = "PalmLauncherRevision" content="version"
              -used to provide a version number associated with the web clipping
              app to be shown in the launcher info version screen
   name = "PalmLauncherName" content="name"
              -used to provide a name that the Palm OS Application Launcher
              displays for the web clipping app; also, the WCA Builder
associates
              this name with the application
   name = "PalmLargeIconFilename" content="filename"
              -used in the HTML that is part of the source that will be built by
              the WCA Builder; this icon will be displayed when the user selects
              the Icon view
   name = "PalmSmallIconFilename" content="filename"
              -used in the HTML that is part of the source that will be built by
              the WCA Builder; this icon will be displayed when the user selects
              the List view
   name = "PalmPostEncoding" content="encoding"
              -used in the HTML that is part of the source that will be build by
              the WCA Builder; used to specify the encoding used for web
              clippings
   name = "PalmPQAVersion" content="value"
              -used in the HTML that is part of the source that will be build by
              the WCA Builder; content contains of value of either "1" or
              "2";used to specify the version of the Clipper that is required to
              run the application
```

```
   name = "PalmPQABitDepth" content="value"
             -used in the HTML that is part of the source that will be build by
             the WCA Builder; used to specify the maximum bit depth for images
             on web clipping pages

-->

<!ELEMENT META - O EMPTY -- Generic Metainformation -->
<!ATTLIST META
   http-equiv    NAME    #IMPLIED   -- HTTP response header name --
   name          NAME    #IMPLIED   -- metainformation name --
   content       CDATA   #REQUIRED  -- associated information
                                       (special strings &date and &time can used
                                        to include the date and time) --
   >

<!--========================== Document Structure ========================-->

<!ENTITY % version.attr "VERSION CDATA #FIXED 'HTML.Version;'">

<![ %HTML.Deprecated [
   <!ENTITY % html.content "HEAD, BODY, PLAINTEXT?">
]]>

<!ENTITY % html.content "HEAD, BODY">

<!ELEMENT HTML O O (%html.content)>
<!ATTLIST HTML
   %version.attr;
   >

<!--================= Character Entities for ISO Latin-1 ====================-->

<!--
     (C) International Organization for Standardization 1986
     Permission to copy in any form is granted for use with
     conforming SGML systems and applications as defined in
     ISO 8879, provided this notice is included in all copies.
     This has been extended for use with HTML to cover the full
     set of codes in the range 160-255 decimal.
-->

<!-- Character entity set. Typical invocation:
     <!ENTITY % ISOlat1 PUBLIC
       "ISO 8879-1986//ENTITIES Added Latin 1//EN//HTML">
     %ISOlat1;
-->
    <!ENTITY nbsp   CDATA " " -- no-break space -->
```

```
<!ENTITY iexcl  CDATA "&#161;" -- inverted exclamation mark -->
<!ENTITY cent   CDATA "&#162;" -- cent sign -->
<!ENTITY pound  CDATA "&#163;" -- pound sterling sign -->
<!ENTITY curren CDATA "&#164;" -- general currency sign -->
<!ENTITY yen    CDATA "&#165;" -- yen sign -->
<!ENTITY brvbar CDATA "&#166;" -- broken (vertical) bar -->
<!ENTITY sect   CDATA "&#167;" -- section sign -->
<!ENTITY uml    CDATA "&#168;" -- umlaut (dieresis) -->
<!ENTITY copy   CDATA "&#169;" -- copyright sign -->
<!ENTITY ordf   CDATA "&#170;" -- ordinal indicator, feminine -->
<!ENTITY laquo  CDATA "&#171;" -- angle quotation mark, left -->
<!ENTITY not    CDATA "&#172;" -- not sign -->
<!ENTITY shy    CDATA "&#173;" -- soft hyphen -->
<!ENTITY reg    CDATA "&#174;" -- registered sign -->
<!ENTITY macr   CDATA "&#175;" -- macron -->
<!ENTITY deg    CDATA "&#176;" -- degree sign -->
<!ENTITY plusmn CDATA "&#177;" -- plus-or-minus sign -->
<!ENTITY sup2   CDATA "&#178;" -- superscript two -->
<!ENTITY sup3   CDATA "&#179;" -- superscript three -->
<!ENTITY acute  CDATA "&#180;" -- acute accent -->
<!ENTITY micro  CDATA "&#181;" -- micro sign -->
<!ENTITY para   CDATA "&#182;" -- pilcrow (paragraph sign) -->
<!ENTITY middot CDATA "&#183;" -- middle dot -->
<!ENTITY cedil  CDATA "&#184;" -- cedilla -->
<!ENTITY sup1   CDATA "&#185;" -- superscript one -->
<!ENTITY ordm   CDATA "&#186;" -- ordinal indicator, masculine -->
<!ENTITY raquo  CDATA "&#187;" -- angle quotation mark, right -->
<!ENTITY frac14 CDATA "&#188;" -- fraction one-quarter -->
<!ENTITY frac12 CDATA "&#189;" -- fraction one-half -->
<!ENTITY frac34 CDATA "&#190;" -- fraction three-quarters -->
<!ENTITY iquest CDATA "&#191;" -- inverted question mark -->
<!ENTITY Agrave CDATA "&#192;" -- capital A, grave accent -->
<!ENTITY Aacute CDATA "&#193;" -- capital A, acute accent -->
<!ENTITY Acirc  CDATA "&#194;" -- capital A, circumflex accent -->
<!ENTITY Atilde CDATA "&#195;" -- capital A, tilde -->
<!ENTITY Auml   CDATA "&#196;" -- capital A, dieresis or umlaut mark -->
<!ENTITY Aring  CDATA "&#197;" -- capital A, ring -->
<!ENTITY AElig  CDATA "&#198;" -- capital AE diphthong(ligature) -->
<!ENTITY Ccedil CDATA "&#199;" -- capital C, cedilla -->
<!ENTITY Egrave CDATA "&#200;" -- capital E, grave accent -->
<!ENTITY Eacute CDATA "&#201;" -- capital E, acute accent -->
<!ENTITY Ecirc  CDATA "&#202;" -- capital E, circumflex accent -->
<!ENTITY Euml   CDATA "&#203;" -- capital E, dieresis or umlaut mark -->
<!ENTITY Igrave CDATA "&#204;" -- capital I, grave accent -->
<!ENTITY Iacute CDATA "&#205;" -- capital I, acute accent -->
<!ENTITY Icirc  CDATA "&#206;" -- capital I, circumflex accent -->
<!ENTITY Iuml   CDATA "&#207;" -- capital I, dieresis or umlaut mark -->
<!ENTITY ETH    CDATA "&#208;" -- capital Eth, Icelandic -->
```

```
<!ENTITY Ntilde CDATA "&#209;" -- capital N, tilde -->
<!ENTITY Ograve CDATA "&#210;" -- capital O, grave accent -->
<!ENTITY Oacute CDATA "&#211;" -- capital O, acute accent -->
<!ENTITY Ocirc  CDATA "&#212;" -- capital O, circumflex accent -->
<!ENTITY Otilde CDATA "&#213;" -- capital O, tilde -->
<!ENTITY Ouml   CDATA "&#214;" -- capital O, dieresis or umlaut mark -->
<!ENTITY times  CDATA "&#215;" -- multiply sign -->
<!ENTITY Oslash CDATA "&#216;" -- capital O, slash -->
<!ENTITY Ugrave CDATA "&#217;" -- capital U, grave accent -->
<!ENTITY Uacute CDATA "&#218;" -- capital U, acute accent -->
<!ENTITY Ucirc  CDATA "&#219;" -- capital U, circumflex accent -->
<!ENTITY Uuml   CDATA "&#220;" -- capital U, dieresis or umlaut mark -->
<!ENTITY Yacute CDATA "&#221;" -- capital Y, acute accent -->
<!ENTITY THORN  CDATA "&#222;" -- capital THORN, Icelandic -->
<!ENTITY szlig  CDATA "&#223;" -- small sharp s, German (sz ligature) -->
<!ENTITY agrave CDATA "&#224;" -- small a, grave accent -->
<!ENTITY aacute CDATA "&#225;" -- small a, acute accent -->
<!ENTITY acirc  CDATA "&#226;" -- small a, circumflex accent -->
<!ENTITY atilde CDATA "&#227;" -- small a, tilde -->
<!ENTITY auml   CDATA "&#228;" -- small a, dieresis or umlaut mark -->
<!ENTITY aring  CDATA "&#229;" -- small a, ring -->
<!ENTITY aelig  CDATA "&#230;" -- small ae diphthong (ligature) -->
<!ENTITY ccedil CDATA "&#231;" -- small c, cedilla -->
<!ENTITY egrave CDATA "&#232;" -- small e, grave accent -->
<!ENTITY eacute CDATA "&#233;" -- small e, acute accent -->
<!ENTITY ecirc  CDATA "&#234;" -- small e, circumflex accent -->
<!ENTITY euml   CDATA "&#235;" -- small e, dieresis or umlaut mark -->
<!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
<!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
<!ENTITY icirc  CDATA "&#238;" -- small i, circumflex accent -->
<!ENTITY iuml   CDATA "&#239;" -- small i, dieresis or umlaut mark -->
<!ENTITY eth    CDATA "&#240;" -- small eth, Icelandic -->
<!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
<!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
<!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
<!ENTITY ocirc  CDATA "&#244;" -- small o, circumflex accent -->
<!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
<!ENTITY ouml   CDATA "&#246;" -- small o, dieresis or umlaut mark -->
<!ENTITY divide CDATA "&#247;" -- divide sign -->
<!ENTITY oslash CDATA "&#248;" -- small o, slash -->
<!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
<!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
<!ENTITY ucirc  CDATA "&#251;" -- small u, circumflex accent -->
<!ENTITY uuml   CDATA "&#252;" -- small u, dieresis or umlaut mark -->
<!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
<!ENTITY thorn  CDATA "&#254;" -- small thorn, Icelandic -->

<!ENTITY yuml   CDATA "&#255;" -- small y, dieresis or umlaut mark -->
```

D

Web Clipping Port Assignments

For security reasons, the Palm.Net service limits the number of open ports that you can use in URL specifications in your web clipping applications. Table D.1 shows the available HTTP port numbers.

Table D.1 HTTP port numbers for web clipping applications

Port number
80
81
82
8000
8001
8002
8003
8004
8005
8080
8083
8801

Table D.2 shows the available HTTPS port numbers.

Table D.2 HTTPS port numbers for web clipping applications

Port number
443
8003
1044

E

Palm OS Lz77 Compression Details

This appendix provides detailed information about the Palm OS® implementation of Lz77 compression, which is available in version 4.0 and later of the Palm OS. The compression scheme has the following components:

- an indicator bit
- an index value (11 bits long)
- a length indicator value (4 bits long)

Table E.1 describes these components.

Table E.1 Lz77 Compression in Palm OS

Component	Length	Description
indicator bit	1 bit	If the value of this bit is 1, then the next 8 bits represents a byte value. If the value of this bit is 0, then the next 11 bits represents an index, followed by 4 bits of length information.
index value	11 bits	Points into the data that has already been parsed in the buffer. The buffer is 2048 bytes long.
length information	4 bits	The number of bytes to retrieve from the buffer (starting at the index byte) is 3 + the value of this field. For example: • if the value is 0, retrieve 3 bytes • if the value is 1, retrieve 4 bytes • ... • if the value is 15, retrieve 18 bytes

Note that this implementation yields a maximum compression ratio of 9, based on the following computation:

```
18(bytes) * 8 bits) / 16 bits
```

Index

Symbols

A

B

C

D

E

F

G

H

I

L

M

O

P

Q

S

T

U

V

W

Z