Palm Web Pro 3.5

powered by novarra

Developer Guide

Palm Web Pro 3.5 Developer Guide

Abstract

This Developer Guide contains information for designing Web pages for display on devices using Palm Web Pro 3.5.

Note:

While every effort has been made to ensure technical accuracy, information in this document is subject to change without notice and does not represent a commitment on the part of Novarra, Inc.

Novarra, Inc.
3232 N. Kennicott
Arlington Heights, IL 60004
Tel: (847) 368-7800
Fax: (847) 590-8144
E-mail: info@novarra.com
Web site: www.novarra.com

January 21, 2004

Table of Contents

Palm Web Pro Overview 1

Thank you for choosing Palm Web Pro 3.5, the sophisticated wireless software solution that empowers your consumer and mobile workforce with seamless access to the Internet, your intranet, and corporate applications from Palm's wireless devices.

Internet/intranet content is automatically extended to mobile device users without customization to existing content. This allows Enterprise and Network Providers to implement wireless strategies in the fastest time possible and without the time and cost setbacks required for developing custom applications or re-authoring Web content in another markup language.

Palm Web Pro is a client browser that is installed on Palm devices to serve as an effective user interface for easy navigation across Internet-based content or applications. Palm Web Pro can directly connect to the internet or work with a proxy server, like the Palm Web Pro Server, for a secure end-to-end information delivery solution that provides optimized communication between the wireless device and Internet/intranet content.

Palm Web Pro supports full access to Internet content and supports multiple standards including: HTML, WML, cHTML, xHTML, SSL, JavaScript and Cascading Style Sheets (CSS).

This document will help you ensure that web content is ideally displayed on Palm devices with Palm Web Pro 3.5.

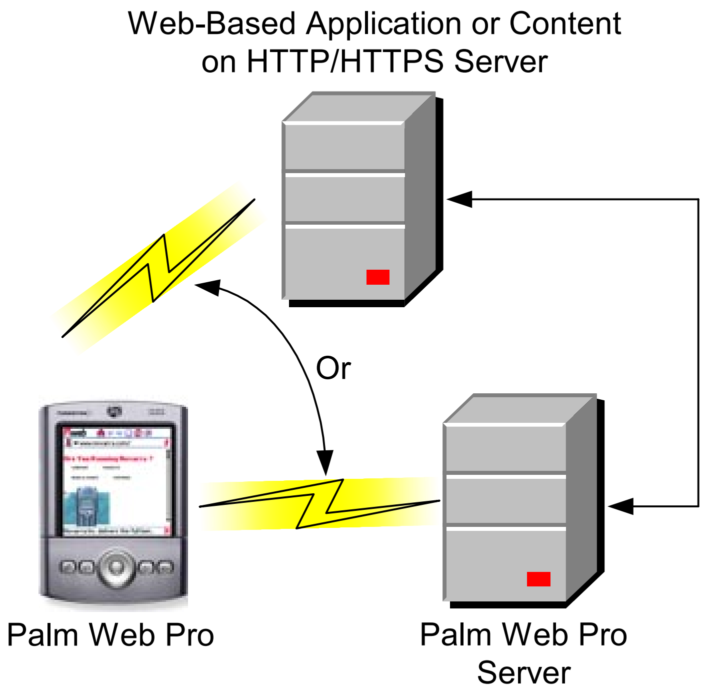
This chapter discusses the following topics:

- "A Look at the Architecture" on page 1-2
- "Detecting Palm Web Pro Access" on page 1-3
- "The Handheld Browsing Experience" on page 1-4
- "Palm Web Pro Support for Internet Standards" on page 1-11
- "Language Support" on page 1-14
- "Contacting Technical Support" on page 1-14
- "For More Information" on page 1-14

1.1 A Look at the Architecture

Palm Web Pro can work effectively with or without a proxy server to access Internet/intranet content and applications.

Palm Web Pro is designed to work best with Novarra’s Palm Web Pro Server to manage network optimization and real-time content adaptation to enhance mobile productivity using the Palm Web Pro. However, Palm Web Pro will work with other proxy servers. Palm Web Pro ensures optimal content presentation and navigation on forthcoming Palm devices.



Palm Web Pro is the default Palm OS handler for most types of content, identified by their MIME types: text/html, etc. Palm Web Pro registers for the http and https schemes to become the default handler for URLs.

1.1.1 Palm Web Pro Server Benefits

Palm Web Pro Server consists of the Adaptive Content Accelerator and the Wireless Connectivity Manager to perform content adaptation, data compression, security and wireless connectivity functions so that users can experience faster download speeds and enjoy content that is adapted for optimal presentation on handheld devices, resulting in an overall more satisfying wireless experience.

Novarra's patent-pending Adaptive Content Accelerator eliminates the need to create a new application or create mirrored sites in different markup languages. It performs real-time adaptation of standard web-based content, including

HTML, JavaScript and WML, for delivery to any wireless device. Key objects including: tables, frames, hypertext links, JavaScript, Cascading Style Sheets and forms are utilized to fully leverage web-based applications on handheld devices. Novarra's Adaptive Content Accelerator also optimizes the presentation of Web content by categorizing it into tiers of information that allow the end user to easily interact with applications from wireless devices.

Novarra's Wireless Connectivity Manager streamlines wireless communications, enables security and manages interaction between an organization's web infrastructure and wireless handheld devices. The Wireless Connectivity Manager controls wireless sessions, protocols and content transmission for efficient and secure communication with wireless devices. It eliminates the need for having to understand “wireless protocols” and simplifies connecting the enterprise to a wireless network. To ensure quick response time, the Wireless Connectivity Manager compresses content, resulting in wireless efficiency and faster downloads. Contact Novarra at www.novarra.com for more information on the Palm Web Pro Server for behind the firewall end to end secure enterprise solutions or our other HTML products.

1.2 Detecting Palm Web Pro Access

Because handhelds typically have smaller screen sizes and other physical characteristics that make them different from desktop computers, you may want to optimize pages specifically for handheld viewing. Your web server can detect Palm Web Pro access requests by looking for the browser’s user-agent string. A simple server-side script can detect the access and display pages that work with the handheld’s configuration.

1.2.1 Proxy/Proxyless User Agent String

Note:

The user agent string for Palm Web Pro is the same for proxy or proxyless.

```
Mozilla/5.0  [en] (PalmOS; U; WebPro/3.5; <manufacturer ID>-<model
ID>)
```

where <manufacturer ID> and <model ID> are replaced with the appropriate ID codes.

Tungsten T3 Example:

```
Mozilla/5.0 [en] (PalmOS; U; WebPro/3.5; Palm-Arz1)
```

Tungsten C Example:

```
Mozilla/5.0 [en] (PalmOS; U; WebPro/3.5; palm-MT64)
```

1.3 The Handheld Browsing Experience

Most sites will be rendered automatically by Palm Web Pro in a usable, easy to navigate fashion in full color with clear, crisp images and text. Although Palm Web Pro automatically scales images, sites should minimize the number and size of images to speed download times.

The following sections provide some hints for developing effective web sites under these constraints:

- "Mobility" on page 1-4
- "Connectivity" on page 1-5
- "Caching" on page 1-5
- "Screen Size and Resolution" on page 1-6
- "Layout Views" on page 1-7
- "Active Graffiti" on page 1-7
- "Landscape Mode" on page 1-7
- "Images and Color" on page 1-7
- "Image Maps" on page 1-8
- "Navigation and Scrolling" on page 1-8
- "Tables" on page 1-8
- "Frames." on page 1-9
- "Multimedia Objects" on page 1-9
- "Forms" on page 1-9
- "Security" on page 1-9
- "Interaction with Other Applications" on page 1-10

1.3.1 Mobility

Mobile browsing is inherently different from the desktop browsing experience. Palm Web Pro has many mobile friendly features, like off-line access, that enhance the mobile experience. Palm Web Pro allows you to save, and later access, web pages even when you are not online. Features to accommodate the lack of a mouse, including support for the 5-way Navigator, are also included.

1.3.2 Connectivity

Palm devices work over wireless networks including Bluetooth, 802.11, IR and GSM/GPRS. Typical data rates supported by wireless carriers are between 19.6 kbps and 56 kbps. Mobile users generally need to get to information quickly (especially when they're charged by the minute), so be mindful of download times. You should design your website to issue and return pages to users as quickly as possible. Avoid large graphics or other data-heavy content, and consider distributing information in smaller chunks.

1.3.3 Caching

Consider taking advantage of Palm Web Pro's local caching features to minimize the amount of downloaded data. For instance, if the home page of your web site rarely changes, you can set a large max-age for that page. This setting will allow Palm Web Pro to load the locally cached version of the page.

Palm Web Pro lets you control caching with the following HTTP/1.1 header elements and directives. You may also send any of these elements in HTML META tag format, which will override the HTTP header elements.

See the HTTP/1.1 specification at http://www.w3c.org for more information about each header element and directive.

1.3.3.1 Cache-Control: max-age

The content is fresh for the specified number of seconds. This directive takes priority over the Expires element.

META tag example:

```
<meta http-equiv="max-age" content="300">
```

1.3.3.2 Cache-Control: no-cache/no-store

The content expires when received and will not be used without further revalidation with the web server. The content is never served from cache.

For compatibility with HTTP/1.0, Palm Web Pro also supports the `Pragma: no-cache` directive.

META tag example:

```
<meta http-equiv="no-cache" content="true">
<meta http-equiv=”no-store” content=”true”>
```

1.3.3.3 Cache-Control: must-revalidate

The content will be revalidated with the web server every time it is accessed.

META tag example:

```
<meta http-equiv="must-revalidate" content="true">
```

1.3.3.4 Expires:"GMT-TIME-IN-ASCII"

The content expires at the specified time.

META tag example:

```
<meta http-equiv="expires" content="Fri, 02 Mar 2003 01:41:31
GMT">
```

1.3.3.5 Last-Modified:"GMT-TIME-IN-ASCII"

This element is used to determine the freshness of expired content. This information is sent to the web server. If the content is still fresh, the server returns a 304 (Not Modified) response and Palm Web Pro displays the cached content. If it is not fresh, the server sends the fresh content.

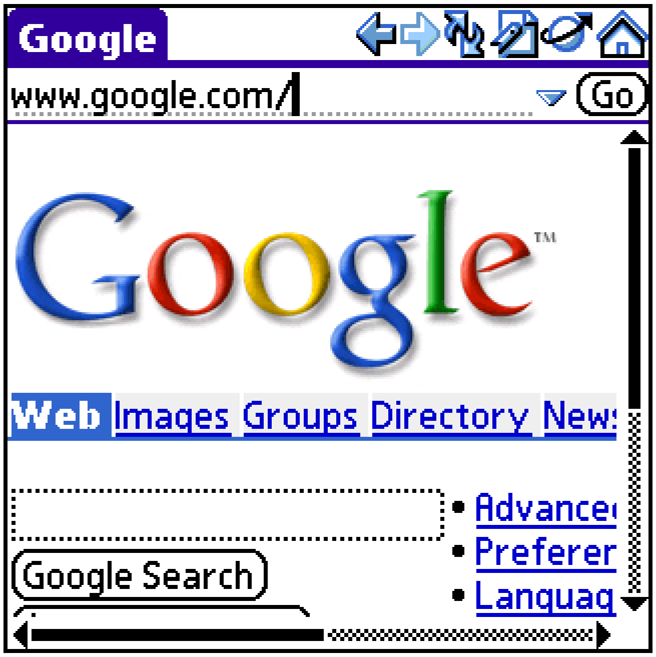
META tag example:

```
<meta http-equiv="last-modified" content="Fri, 02 Mar 2001
01:41:31 GMT">
```

1.3.4 Screen Size and Resolution

Palm Web Pro is designed for resolutions of 320 x 480, 320 x 320 and 160 x 160 pixels. Palm Web Pro will scale your images and text to fit this screen size.

Other screen elements, such as the Palm Web Pro title bar and scroll bars, take up some of this limited space. The figure below shows an example 320 x 320 screen where the vertical and horizontal scroll bars are present. Screen resolution for actual content area in the example below is 304 x 260.



For devices with a 320 x 480 screen, the actual content area depends on the display mode:

Note:

Certain devices may use additional pixels to display a status bar making the actual content area a bit smaller then listed below.

Landscape - 464 x 260

Portrait - 304 x 420

The above dimensions can be used in optimizing high resolution image sizes. Keep in mind, however, that the Palm OS® still calculates text sizes at low resolution (for backward compatibility).

1.3.4.1 Full Screen View

In Full Screen View, the menu and URL fields are removed giving a larger viewing area.

1.3.5 Layout Views

Palm Web Pro provides two layouts:

- Handheld View - Horizontal scrolling is minimized. Font size is maintained as normal (user-set). Text is wrapped and images are scaled to fit within the screen width.
- Normal View - The webpage is display as it would appear on a desktop browser. Horizontal scrolling will be used extensively.

1.3.6 Active Graffiti

Palm Web Pro occupies screen space made available on devices that have the ability to make the graffiti area disappear giving more room for viewing.

1.3.7 Landscape Mode

Palm Web Pro supports switching from portrait to landscape mode for devices with screen resolutions of 320 x 480.

1.3.8 Images and Color

Palm Web Pro scales images and graphics to fit on the small screen. However, you may want to design a handheld-specific page that uses smaller images. Reducing the size and number of colors will also help reduce the download times for the user.

The image types that are supported are:

- GIF
- Animated GIF
- JPEG
- PNG

- BMP (supported on the server but not proxyless)
- WBMP

Tip:

Avoid using spacer images to align content. Spacer images may get scaled or dropped by the Palm Web Pro server, resulting in misaligned content.

1.3.9 Image Maps

Image maps are supported. On a desktop browser, moving the mouse over the usable areas of the image often shows a small tip window to indicate what that area does. Since there are no "mouse over" events when using a stylus, these are not shown by Palm Web Pro.

1.3.10 Navigation and Scrolling

Consider how users will navigate your site using controls common to Palm OS handhelds. Remember that users will be using a stylus or 5-Way Navigator instead of a mouse, which means that active feedback involving rollovers or “hovering” is significantly less useful. They will often use their handhelds one-handed, tapping on your links with the stylus, so your target sizes may need to be increased to compensate for reduced dexterity.

Palm Web Pro, in Handheld View, will automatically render content vertically to minimize horizontal scrolling.

1.3.11 Tables

Although Web designers commonly employ tables for page layout control on desktop browsers, their intended function has always been for organizing tabular data. Where possible, minimize the use of tables for organizing page layout.

Tables are supported on Palm Web Pro. In Handheld View, single tables will display similar to a desktop browser which means the horizontal scroll bar may appear. Nested tables are unrolled from left to right, top to bottom.

Tip:

- Minimize the use of tables for organizing page layout. Use tables primarily for organizing data.
- Minimize the use of nested tables. Nested tables are not particularly well suited for small screen, and will be unrolled by Palm Web Pro server. When a table is shown in Handheld View, content may be displayed in non-tabular for to improve visibility.

1.3.12 Frames.

Palm Web Pro can display pages that are divided into frames.

In Handheld View, frames do not appear on Palm Web Pro as they do on a desktop browser. Frames are rendered vertically left to right, top to bottom.

Tip:

Specify the frames from top to bottom in the frameset. Frames are displayed in Palm Web Pro based on the order they are encountered in the HTML source.

1.3.13 Multimedia Objects

Palm Web Pro ignores **APPLET**, **EMBED**, and **OBJECT** tags and their content.

Tip:

To make navigation consistent between desktop browsers and Palm Web Pro, do not rely solely on multimedia elements, such as Flash movies for navigation.

1.3.14 Forms

Palm Web Pro supports forms and all standard HTML input controls.

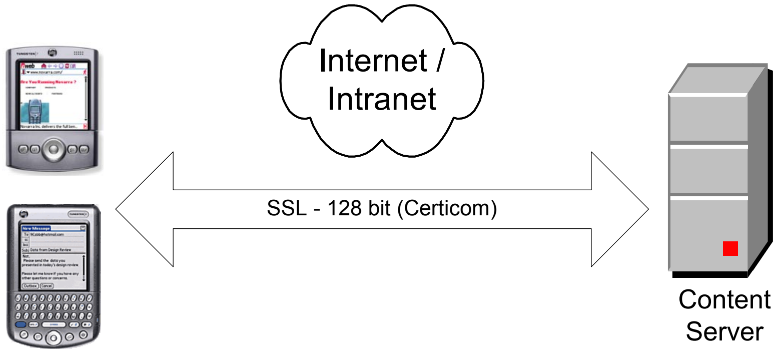
1.3.15 Cookies

A cookie stores user information that a Web server can access. Typical uses are for shopping-cart applications or to store buying preference information. Most cookies have lifetimes associated with them.

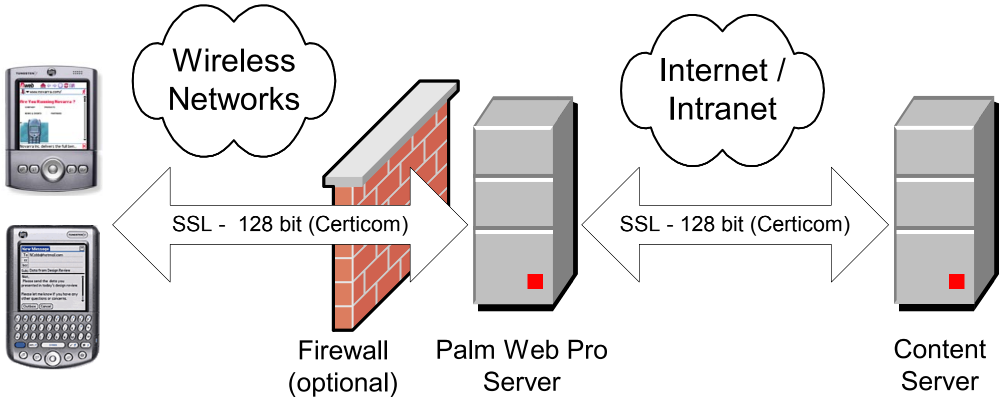
Palm Web Pro supports persistent and session based cookies.

1.3.16 Security

To provide end-to-end security between Palm Web Pro and Internet/intranet content, Palm Web Pro uses Secure Sockets Layer (SSL). SSL enables secure communications to servers that support SSL. This secure communication allows encryption /decryption of sensitive data transmitted to a secure site.



- or -



1.3.17 Interaction with Other Applications

- Palm Web Pro is able to download files from the Internet/intranet. If an application on the device supports the downloaded file type, Palm Web Pro will send the file to the application via the Exchange Manager.
- Palm Web Pro is registered with the Exchange Manager for the http and https schemes. Applications like VersaMail will allow the user to tap on the link to launch Palm Web Pro and load the page.
- Tapping on a `mailto:` link will launch VersaMail
- Tapping on a `phoneto:` link will launch the Dialer

For more information, see Chapter 7, “Integrating Applications with Palm Web Pro.”

1.4 Palm Web Pro Support for Internet Standards

Palm Web Pro supports common Internet standards as described in the following sections.

1.4.1 HTML Support

Palm Web Pro supports the WC3 HTML 4.01 standard.

1.4.2 Scripting Support

Tip:

When using Palm Web Pro Server, avoid using the **`onchange`** attribute on controls. This attribute may result in additional traffic between the device and the server.

Palm Web Pro supports the ECMAScript (ECMA-262) standard, which is a unified standard based on JavaScript. It incorporates most aspects of the W3C Document Object Model (DOM). Most web pages designed with JavaScript 1.3 or JScript 5.0 will be compatible with Palm Web Pro. Test your scripts in Palm Web Pro to ensure compatibility.

If your script includes an unsupported function, the script will continue on to the next function. Most script errors will fail silently. However, the window.open function will fail with an error if the command is the result of a user action, such as clicking on a link that opens a popup window. Other situations, such as calling the **`window.open`** function **`ONLOAD`**, will result in a silent failure. Palm Web Pro does not support other client-side scripting languages such as VBScript. If the user unchecks the “Enable JavaScript” box in the Palm Web Pro options screen (Options → Preferences → Advanced), then Palm Web Pro will render the **`NOSCRIPT`** content.

1.4.3 CSS1

Palm Web Pro is CSS 1 compliant. The following items are supported:

- External Style Sheets
- Imported Style Sheets
- Embedded and Inline Styles
- Inheritance
- Cascading

1.4.4 Other Standards

Palm Web Pro does not support the following desktop features:

- Applets
- VBScript
- ActiveXControls

1.5 Special Features

1.5.1 Support for LocalHost Access

Palm Web Pro supports the ability to access a LocalHost (127.0.0.1) server socket for http:// traffic. This allows the user to utilize an on-device micro web server running in the background. To see this, go to Options→Preferences→ Advanced and set 127.0.0.1 as the proxy server address and 80 as the port.

1.5.2 Support for file:// Scheme

Palm Web Pro supports the file:// scheme for expansion card access including all supported content types - html, images, etc. and also supports relative links.

The user is able to open a file on the SD card from within the browser and download a file to the SD card.

The user can load HTML, jpeg and gif files from their desktop computer to the SD card. Then they can put the SD card in the Palm device and view the files in the browser.

HTML files on the SD card can have relative links to locate other images on the SD card.

1.5.3 Palm: and PalmCall: Support

Palm Web Pro supports the Palm: and PalmCall: URL schemes to run another application on the Palm device. With PalmCall:, Palm Web Pro gets suspended in the background while the other application runs. When the application exits, the browser reactivates in the same state it was in before performing the sub-launch. With Palm:, Palm Web Pro exits and the other application runs as the main application. For more information, see section 7.1, “Launching an Application from a Web Page” on page 7-1.

1.5.4 File Download Capabilities

Palm Web Pro is able to download files from the web - .prc, .pdb, .pqa, .doc, .txt, .mp3, .pdf, etc. files. The .pqa and .prc files are shown in the application launcher after download.

The user can download any type of file to an SD card if it is available. It is added to the Web Pro folder on the SD card.

The user is alerted before the download is begun; during the download, the user is shown a status bar with percentage of download completed and after the download is completed, the user is shown a message saying so.

Palm Web Pro is able to download any file type that is registered with the Exchange Manager.

In case the application on the device (such as application to view documents and spreadsheets) uses a proprietary format to view .doc or .pdf files, Palm Web Pro will pass the downloaded files as-is.

If the file type is not supported by any application registered with the Exchange Manager, the Palm Web Pro displays an error. The file can only be downloaded to an SD card.

Note that html, .gif and .jpg are always shown within Palm Web Pro rather than downloaded to an SD card or another application.

1.5.5 Certificates

Palm Web Pro gives the user an option to visit the secure website even if:

- The security certificate is not valid.
- The issuing authority for the security certificate is not trusted by the user.
- The name of the security certificate does not match the name of the site.

The browser gives the user a warning and ask if he/she wants to proceed. The user can view the details of the certificate before choosing whether to proceed.

For a list of the supported certificates for the proxy and proxyless browser, see Appendix A, “Supported Certificates.”

1.5.6 Pluggable Scheme Handlers

The pluggable scheme handler (PSH) feature gives third party developers the ability to extend the client to handle protocols that the client does not natively support. For example the FTP (File Transfer Protocol) is not natively supported but could be added with a PSH.

For more information see Chapter 6, “Pluggable Scheme Handlers.”

1.6 Language Support

Palm Web Pro supports the following languages:

- English
- French
- German
- Italian
- Spanish
- Brazilian Portuguese

1.7 Contacting Technical Support

Visit the appropriate Palm Customer Support Web site for technical questions and other technical-related issues.

US and Canada: http://www.palm.com/support/contact/phone_support.html

International: http://www.palm.com/support/globalsupport.html

When contacting technical support, please be prepared to provide the following information:

- General contact information, including name, company, e-mail, and phone number
- Type of mobile device in use
- Type and version of the operating system
- A brief description of the problem and the steps necessary to recreate it
- Specific error messages

1.8 For More Information

Refer to the following for more information:

- Chapter 2, “HTML, XHTML & cHTML Support Reference.”
- Chapter 4, “Style Sheet Support Reference.”
- Chapter 5, “JavaScript Support Reference.”
- Chapter 6, “Pluggable Scheme Handlers.”
- Chapter 7, “Integrating Applications with Palm Web Pro.”
- Appendix A, “Supported Certificates.”

HTML, XHTML & cHTML Support Reference 2

The tables in this section will help you learn the differences between the HTML elements defined in the W3C HTML 4.01 specification (see http://www.w3.org/TR/REC-html40/) and those supported by Palm Web Pro. Remember that even when supported, the element's presentation in Palm Web Pro may differ from common desktop browsers.

Note:

The information listed in the following tables are also applicable to XHTML and cHTML.

Information on the different elements can be found in the following sections:

- "Head Elements" on page 2-2
- "Body and Text Elements" on page 2-3
- "Font and Style Elements" on page 2-5
- "List Elements" on page 2-7
- "Table and Frame Elements" on page 2-8
- "Link Elements" on page 2-11
- "Images and Other Object Elements" on page 2-12
- "Form Elements" on page 2-13
- "Other Elements" on page 2-17

Palm Web Pro 3.5 Usage Key

- **Y (Yes)** The tag and its contents are processed.
- **N (No)** The tag and its contents are skipped.
- **I (Included)** The children of the tag are processed.
- **P (Partial)** The tag and its contents are utilized. However, the standard is not followed to the letter and the presentation is affected.

Palm Web Pro 3.5 Notes Key
1. This tag or attribute is not fully supported when using the Palm Web Pro Server. 2. When using Handheld View or when using the Palm Web Pro Server, content targeted to frames is shown in isolation to improve the user experience on a wireless handheld device. 3. Cellspacing and cellpadding - supported, but the browser may reduce the value as needed. 4. Width - relative width is not supported in Handheld View only. The browser will use an absolute value as "suggested width". 5. If the actual image size is different from the supplied height/width attributes, the image will not be scaled. If the image is larger than the supplied size, it will be cropped. Conversely, if the image is smaller, the extra space will remain blank. Images larger than the screen in any dimension will be scaled to fit the screen regardless of the supplied width and height attributes.

2.1 Head Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
HTML		Y	Y	
	version	Y	Y	
	xmins	N	Y	
HEAD		Y	Y	
	profile	Y	Y	
TITLE		Y	Y	
META		Y	Y	
	http-equiv	Y	Y	
	name	Y	N	
	content	Y	N	
	scheme	Y	N	

2.2 Body and Text Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
BODY		Y	Y	
	onload	Y	Y	
	onunload	Y	N	
	background	Y	Y	1
	text	Y	Y	1
	link	Y	Y	
	vlink	Y	Y	
	alink	Y	N	
	bgcolor	Y	Y	1
	bgproperties	N	N	
	bottommargin	N	N	
	leftmargin	N	N	
	rightmargin	N	N	
	scroll	N	N	
	topmargin	N	N	
DIV, SPAN		Y	Y	
	align	Y	Y	1
H1 – H6		Y	I	
	align	Y	Y	1
ADDRESS		Y	Y	
EME		Y	Y	
STRONG		Y	Y	
DFN		Y	Y	
CODE		Y	Y	
SAMP		Y	Y	
KBD		Y	Y	
VAR		Y	Y	
CITE		Y	Y	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
ABBR		Y	Y	
ACRONYM		Y	Y	
BLOCKQUOTE, Q		Y	I	
	cite	Y	N	
SUB		Y	I	
SUP		Y	I	
P		Y	Y	
	align	Y	Y	1
BR		Y	Y	
	clear	Y	N	
PRE		Y	I	
	width	Y	N	
INS, DEL		Y	I	
	cite	Y	Y	
	datetime	Y	Y	
BLINK		N	N	
MARQUEE		N	N	

2.3 Font and Style Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
STYLE		Y	Y	
	type	Y	Y	
	media	Y	Y	
	title	Y	Y	
CENTER		Y	Y	1
TT		Y	Y	
I		Y	Y	
B		Y	Y	
BIG		Y	Y	
SMALL		Y	Y	
STRIKE		Y	Y	
S		Y	Y	
U		Y	Y	
FONT		Y	Y	
	size	Y	Y	
	color	Y	Y	
	face	Y	N	
BASEFONT		Y	Y	
	id	Y	I	
	size	Y	Y	
	color	Y	Y	
	face	Y	N	
HR		Y	Y	
	align	Y	Y	1
	noshade	Y	Y	
	size	Y	N	
	width	Y	Y	1
	behavior	N	N	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
	bgcolor	N	N	
	direction	N	N	
	height	N	N	
	hspace	N	N	
	loop	N	N	
	scrollamount	N	N	
	scrolldelay	N	N	
	vspace	N	N	
	width	N	N	
BDO		Y	N	

2.4 List Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
UL		Y	Y	
	type	Y	N	
	compact	Y	N	
OL		Y	Y	
	type	Y	N	
	start	Y	Y	
	compact	Y	N	
LI		Y	Y	
	type	Y	N	
	value	Y	N	
	compact	Y	N	
DIR		Y	Y	
	compact	Y	N	
MENU		Y	Y	
	compact	Y	N	
DL		Y	Y	
	compact	Y	N	
DT		Y	Y	
DD		Y	Y	

2.5 Table and Frame Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
TABLE		Y	Y	
	summary	Y	N	
	width	Y	Y	4
	border	Y	Y	
	frame	Y	Y	
	rules	Y	N	
	cellspacing	Y	Y	3
	cellpadding	Y	Y	3
	align	Y	Y	1
	bgcolor	Y	Y	
	background	N	N	
	bordercolor	N	N	
CAPTION		Y	Y	
	align	Y	Y	1
THEAD, TFOOT, TBODY		Y	Y	
	align	Y	Y	1
	char	Y	N	
	charoff	Y	N	
	valign	Y	N	
	bgcolor	N	N	
COLGROUP		Y	Y	1
	span	Y	Y	1
	width	Y	Y	1
	align	Y	Y	1
	char	Y	N	
	charoff	Y	N	
	valign	Y	N	
	bgcolor	N	N	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
COL		Y	Y	1
	span	Y	Y	1
	width	Y	Y	1
	align	Y	Y	1
	char	Y	N	
	charoff	Y	N	
	valign	Y	N	
	bgcolor	N	N	
TR		Y	Y	
	bgcolor	Y	Y	
	align	Y	Y	1
	char	Y	N	
	charoff	Y	N	
	valign	Y	N	
	background	N	N	
	bordercolor	N	N	
TH, TD		Y	Y	
	abbr	Y	Y	1
	axis	Y	N	
	headers	Y	N	
	scope	Y	N	
	rowspan	Y	Y	
	colspan	Y	Y	
	align	Y	Y	1
	char	Y	N	
	charoff	Y	N	
	valign	Y	N	
	nowrap	Y	Y	
	width	Y	Y	4
	height	Y	Y	1
	bgcolor	Y	Y	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
	background	N	N	
	bordercolor	N	N	
FRAMESET		Y	Y	2
FRAME		Y	Y	2
NOFRAMES		Y	Y	
IFRAME		Y	Y	2

2.6 Link Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
A		Y	Y	
	charset	Y	N	
	type	Y	N	
	name	Y	Y	
	href	Y	Y	
	hreflang	Y	N	
	rel	Y	N	
	rev	Y	N	
	accesskey	Y	N	
	shape	Y	N	
	coords	Y	N	
	tabindex	Y	N	
	onfocus	Y	N	
	onblur	Y	N	
	target	Y	Y	2
LINK		Y	Y	
	charset	Y	N	
	href	Y	Y	
	hreflang	Y	N	
	type	Y	Y	1
	rel	Y	N	
	rev	Y	N	
	media	Y	N	
BASE		Y	Y	
	href	Y	Y	
	target	Y	Y	2

2.7 Images and Other Object Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
IMG		Y	Y	
	src	Y	Y	
	alt	Y	Y	
	longdesc	Y	N	
	name	Y	N	
	height	Y	Y	5
	width	Y	Y	5
	usemap	Y	Y	
	ismap	Y	N	
	align	Y	Y	1
	border	Y	Y	
	hspace	Y	Y	
	vspace	Y	Y	
OBJECT		Y	N	
PARAM		Y	N	
APPLET		Y	N	
MAP		Y	Y	
AREA		Y	Y	

2.8 Form Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
FORM		Y	Y	
	action	Y	Y	
	method	Y	Y	
	enctype	Y	N	
	accept	Y	N	
	name	Y	Y	
	onsubmit	Y	Y	
	onreset	Y	Y	1
	accept-charset	Y	N	
	target	Y	Y	2
INPUT		Y	Y	
	disabled	Y	Y	
	tabindex	Y	N	
	accesskey	Y	N	
	onfocus	Y	N	
	onblur	Y	N	
	onselect	Y	N	
	onchange	Y	Y	
	type = button	Y	Y	
	name	Y	Y	
	value	Y	Y	
	type = checkbox	Y	Y	
	checked	Y	Y	
	name	Y	Y	
	value	Y	Y	
	type = file	Y	N	
	type = hidden	Y	Y	
	name	Y	Y	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
	value	Y	Y	
	type = image	Y	Y	
	align	Y	Y	1
	alt	Y	Y	
	border	N	N	
	height	N	Y	1
	ismap	Y	N	
	name	Y	Y	
	src	Y	Y	
	usemap	Y	Y	1
	value	Y	Y	
	width	N	Y	1
	type = password	Y	Y	
	maxlength	Y	Y	
	name	Y	Y	
	size	Y	Y	
	value	Y	Y	
	readonly	Y	Y	
	type = radio	Y	Y	
	checked	Y	Y	
	name	Y	Y	
	value	Y	Y	
	type = reset	Y	Y	
	name	Y	Y	
	value	Y	Y	
	type = submit	Y	Y	
	name	Y	Y	
	value	Y	Y	
	type = text	Y	Y	
	istyle	N	N	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
	maxlength	Y	Y	
	name	Y	Y	
	size	Y	Y	
	value	Y	Y	
	readonly	Y	Y	
BUTTON		Y	Y	
	name	Y	Y	
	value	Y	Y	
	type	Y	Y	
	disabled	Y	N	
	tabindex	Y	N	
	accesskey	Y	N	
	onfocus	Y	N	
	onblur	Y	N	
SELECT		Y	Y	
	name	Y	Y	
	size	Y	Y	
	multiple	Y	Y	
	disabled	Y	N	
	tabindex	Y	N	
	onfocus	Y	N	
	onblur	Y	N	
	onchange	Y	Y	
	accesskey	N	N	
OPTGROUP		Y	N	
OPTION		Y	Y	
	selected	Y	Y	
	disabled	Y	N	
	label	Y	Y	
	value	Y	Y	
TEXTAREA		Y	Y	

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
	name	Y	Y	
	rows	Y	Y	
	cols	Y	N	
	disabled	Y	N	
	readonly	Y	Y	
	tabindex	Y	N	
	accesskey	Y	N	
	onfocus	Y	N	
	onblur	Y	N	
	onselect	Y	N	
	onchange	Y	Y	1

2.9 Other Elements

Tag	Attribute	HTML 4.01	Palm Web Pro 3.5	Notes
ISINDEX		Y	N	
LABEL		Y	N	
FIELDSET		Y	N	
LEGEND		Y	N	
SCRIPT		Y	Y	
NOSCRIPT		Y	Y	

WML Support Reference 3

3.1 WML Processing and Support

Note:

WML is not supported in the proxyless version of Palm Web Pro.

Palm Web Pro 3.5 Usage Key

- **Y (Yes)** The tag and its contents are processed.
- **N (No)** The tag and its contents are skipped.
- **I (Included)** The children of the tag are processed.
- **P (Partial)** The tag and its contents are utilized. However, the standard is not followed to the letter and the presentation is affected.

Palm Web Pro 3.5 Notes Key

1. Device “Back” key not supported.
2. Emulates a button for timer expiry.

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
a		Provides a short Form of the Anchor element and is used for bounding without variables.	Y	
access		Describes access control information pertaining to the deck. The decks domain and path attributes specify what other decks can access it. The access type is placed in between the <head> declaration.	N	
anchor		Specifies the head of a link. Can be used around any formatted text. Each Anchor must be assigned a task: (go, prev, refresh)	Y	
b		Text formatting render with Bold font.	Y	
big		Text formatting render with Large font.	I	
br		The br break element is used to end a current line and begin a new one. **Note:** This tag has a closing slash (/) unlike in HTML.	Y	
card		A card makes up the contents of a deck. There are many types of cards allowing varying manipulation. The entire WML page is contained within card tags (like body in HTML). A single WML page can have more than one card page defined.	Y	
	title	Card information.	Y	
	newcontext	Reinitialize card	Y	
	ordered	Layout of card	N	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
	onenterforward	Used when a card is navigated to by the use of a go task	Y	
	onenterbackward	Used when a card is navigated to by the use of a prev task	P	1
	ontimer		P	2
	id		Y	
	class		N	
do		The do element is used for traversing WML pages and cards. This can be links and form submission.	Y	
	type=”{accept, prev, help, reset, options, delete, unknown]”		Y	
	label	Textual string	Y	
	name	Name of do event optional	Y	
	optional		N	
em		Render text with emphasis.	Y	
fieldset		Used for grouping text and related input fields. Used by browser for optimizing layout.	I	
	title	Label for fieldset.	N	
go		Used inside anchor and do tags for destination setting. The go task defines navigation to a URL.	Y	
	href	Destination URL.	Y	
	method=”[get, post]”	http submission method.	Y	
	sendreferer		N	
	accept-charset		N	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
head		Contains information relevant to the deck (WML page) as a whole. Must include either an access or meta element.	Y	
i		Text contained within the tags is rendered in italics.	Y	
img		Used to place images. The same layout is used as normal text	Y	
	alt	Alternative textual representation if graphic is not supported.	Y	
	src	URL of image.	Y	
	vspace	The amount of white space to be inserted around the image.	N	
	hspace	The amount of white space to be inserted around the image.	N	
	align=”[top, middle, bottom]”		N	
	height		N	
	width		N	
input		The Input element allows the user to enter text assigned to a specific variable.	Y	
	name	Name of the variable used to hold input value.	Y	
	title	label	Y	
	value	Default value of named attribute.	Y	
	type	Text or password.	Y	
	format=”A”	Uppercase alphabetic or punctuation.	N	
	format=”a”	Lowercase alphabetic or punctuation.	N	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
	format=”N”	Numeric character.	N	
	format=”X”	Symbolic, numeric or Uppercase alphabetic character (Non Changeable to lowercase).	N	
	format=”M”	Symbolic, numeric or uppercase alphabetic character (changeable to lowercase).	N	
	format=”m”	Symbolic, numeric or lowercase alphabetic character (changeable to uppercase).	N	
	emptyok	Boolean specifying weather field may be left blank.	N	
	size	Width in chars of text input area.	N	
	maxlength	Maximum number of allowed inputted characters	N	
	tabindex	Tabbing position of current element	N	
meta		Contains meta data. Used for keywords, content-type, author, etc. Information relating to the deck.	Y	
noop		The no operation task. A task element that instructs the device to do nothing.	Y	
onevent		The onevent element associates a state transition. Each transition is associated with a type and a task.	Y	
	type=”onpick”		Y	
	type=”onenterforward”		Y	
	type=”onenterbackward”		P	
	type=”ontimer”		P	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
ontimer		Contained in the card element to specify a URL to move to after a time set by timer.	N	
optgroup		The optgroup element allows the grouping of multiple options within a card. Used for grouping Option elements into a single selection.	I	
option		The option element specifies a particular choice within the <select> element. It offers either a radio or checkbox selection. Radio is available when grouped with an OptGroup otherwise standard checkbox action is supplied.	Y	
	title	Label that identifies option.	Y	
	value	Value to assign to variable defined in select.	Y	
	onpick	URL to open if used.	Y	
p		Text formatting establish paragraph.	Y	
	align=”[left, right, center]”		Y	
	mode=”[wrap/nowrap]”		N	
postfield		Defines name and value pairs that are passed to the HTTP server.	Y	
prev		Used to navigate to previous URL in stack	Y	
refresh		Used for refreshing values when a refresh is called by the browser.	Y	
select		Creates a list of input elements which are grouped together.	Y	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
	title	label	Y	
	name	Name of the variable that the devices stores the value associated to the option.	Y	
	value		Y	
	iname		Y	
	ivalue		Y	
	multiple	Boolean used to dictate if having multiple items.	Y	
	Default	Default value of variable specified in name attribute.	N	
	tabindex	Tabbing position of current element.	N	
setvar		Sets a variable to a specified value.	Y	
small		Text formatting render with small font.	I	
strong		Text formatting render with strong emphasis.	Y	
table		Table element is used with tr and td to define aligned columns for text or images (same as HTML tables in a WML page).	Y	
	title	Table information	Y	
	align=”[left(l), right(r), center(c)]”		N	
	columns	number	N	
	id		Y	
	class		Y	
td		td element is used to hold a single table column. Used by table to assemble table data cells.	Y	

Tag	Attribute	Description	Palm Web Pro 3.5	Notes
template		The template element declares a template for cards in the deck. It is used in conjunction with types and can be seen as a procedural navigational method.	Y	
	type=”onenterforward”		Y	
	type=”onenterbackwards”		P	
	type=”ontimer”		P	
timer		Sets the time in milliseconds before activating the ontimer event.	N	
tr		Used by table to assemble table data rows.	Y	
u		Text contained within the tags is underlined	I	
wml		Defines the Deck and encloses all the information and cards in it.	Y	

Style Sheet Support Reference 4

Palm Web Pro supports various aspects of the CSS1 and CSS2 standards. The tables in the following sections compare the Web Browser software’s CSS support with the CSS1 and CSS2 standards:

Palm Web Pro 3.5 Usage Key

- **Y (Yes)** This property or value is supported and is processed.
- **N (No)** This property or value is not supported and is skipped.

Palm Web Pro 3.5 Notes Key

1. This value or pattern is not supported when using the Palm Web Pro Server.

4.1 Alignment

These properties are not supported

4.2 Auto-Generated Characters

These properties are not supported

4.3 Audio Replay

These properties are not supported

4.4 Boxes

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
border		Y	Y	Y	
	'border-width' || 'border-style' || <color>	Y	Y	Y	
	inherit	N	Y	Y	
border-color		Y	Y	Y	1
	<color>{1,4}	Y	Y	Y	1
	transparent	Y	Y	Y	1
	inherit	N	Y	Y	1
border-style		Y	Y	Y	1
	none	Y	Y	Y	1
	hidden	N	Y	Y	1
	dotted	Y	Y	N	
	dashed	Y	Y	N	
	solid	Y	Y	Y	1
	double	Y	Y	N	
	groove	Y	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	ridge	Y	Y	N	
	inset	Y	Y	N	
	outset	Y	Y	N	
	inherit	Y	Y	Y	1
border-top		Y	Y	N	
	'border-top-width' || 'border-top-style' || 'border-top-color'	Y	Y	N	
	inherit	N	Y	N	
border-right		Y	Y	N	
	'border-right-width' || 'border-right-style' || 'border-right-color'	Y	Y	N	
	inherit	N	Y	N	
border-bottom		Y	Y	N	
	'border-bottom-width' || 'border-bottom-style' || 'border-bottom-color'	Y	Y	N	
	inherit	N	Y	N	
border-left		Y	Y	N	
	'border-left-width' || 'border-left-style' || 'border-left-color'	Y	Y	N	
	inherit	N	Y	N	
border-top-color		N	Y	N	
	<color>	N	Y	N	
	transparent	N	Y	N	
	inherit	N	Y	N	
border-right-color		N	Y	N	
	<color>	N	Y	N	
	transparent	N	Y	N	
	inherit	N	Y	N	
border-bottom-color		N	Y	N	
	<color>	N	Y	N	
	transparent	N	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	inherit	N	Y	N	
border-left-color		N	Y	N	
	<color>	N	Y	N	
	transparent	N	Y	N	
	inherit	N	Y	N	
border-top-style		N	Y	N	
	none	N	Y	N	
	hidden	N	Y	N	
	dotted	N	Y	N	
	dashed	N	Y	N	
	solid	N	Y	N	
	double	N	Y	N	
	groove	N	Y	N	
	ridge	N	Y	N	
	inset	N	Y	N	
	outset	N	Y	N	
	inherit	N	Y	N	
border-right-style		N	Y	N	
	none	N	Y	N	
	hidden	N	Y	N	
	dotted	N	Y	N	
	dashed	N	Y	N	
	solid	N	Y	N	
	double	N	Y	N	
	groove	N	Y	N	
	ridge	N	Y	N	
	inset	N	Y	N	
	outset	N	Y	N	
	inherit	N	Y	N	
border-bottom-style		N	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	none	N	Y	N	
	hidden	N	Y	N	
	dotted	N	Y	N	
	dashed	N	Y	N	
	solid	N	Y	N	
	double	N	Y	N	
	groove	N	Y	N	
	ridge	N	Y	N	
	inset	N	Y	N	
	outset	N	Y	N	
	inherit	N	Y	N	
border-left-style		N	Y	N	
	none	N	Y	N	
	hidden	N	Y	N	
	dotted	N	Y	N	
	dashed	N	Y	N	
	solid	N	Y	N	
	double	N	Y	N	
	groove	N	Y	N	
	ridge	N	Y	N	
	inset	N	Y	N	
	outset	N	Y	N	
	inherit	N	Y	N	
border-top-width		Y	Y	N	
	thin	Y	Y	N	
	medium	Y	Y	N	
	thick	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	
border-right-width		Y	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	thin	Y	Y	N	
	medium	Y	Y	N	
	thick	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	
border-bottom-width		Y	Y	N	
	thin	Y	Y	N	
	medium	Y	Y	N	
	thick	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	
border-left-width		Y	Y	N	
	thin	Y	Y	N	
	medium	Y	Y	N	
	thick	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	
border-width		Y	Y	Y	
	<border-width>{1,4}	Y	Y	Y	
	inherit	N	Y	Y	
margin		Y	Y	Y	
	<margin-width>{1,4}	Y	Y	Y	
	inherit	N	Y	Y	
margin-top		Y	Y	Y	
	<length>	Y	Y	Y	
	<percentage>	Y	Y	Y	1
	auto	Y	Y	Y	1
	inherit	N	Y	Y	
margin-right		Y	Y	Y	
	<length>	Y	Y	Y	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	<percentage>	Y	Y	Y	1
	auto	Y	Y	Y	1
	inherit	N	Y	Y	
margin-bottom		Y	Y	Y	
	<length>	Y	Y	Y	
	<percentage>	Y	Y	Y	1
	auto	Y	Y	Y	1
	inherit	N	Y	Y	
margin-left		Y	Y	Y	
	<length>	Y	Y	Y	
	<percentage>	Y	Y	Y	1
	auto	Y	Y	Y	1
	inherit	N	Y	Y	
padding		Y	Y	Y	1
	<padding-width>{1,4}	Y	Y	Y	1
	inherit	N	Y	Y	1
padding-top		Y	Y	Y	1
	<length>	Y	Y	Y	1
	<percentage>	Y	Y	Y	1
	inherit	N	Y	Y	1
padding-right		Y	Y	Y	1
	<length>	Y	Y	Y	1
	<percentage>	Y	Y	Y	1
	inherit	N	Y	Y	1
padding-bottom		Y	Y	Y	1
	<length>	Y	Y	Y	1
	<percentage>	Y	Y	Y	1
	inherit	N	Y	Y	1
padding-left		Y	Y	Y	1
	<length>	Y	Y	Y	1

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	<percentage>	Y	Y	Y	1
	inherit	N	Y	Y	1

4.5 Colors and Backgrounds

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
background		Y	Y	Y	
	'background-color' || 'background-image' || 'background-repeat' || 'background-attachment' || 'background-position'	Y	Y	Y	
	inherit	N	Y	Y	
background-attachment		Y	Y	N	
	scroll	Y	Y	N	
	fixed	Y	Y	N	
	inherit	N	Y	N	
background-color		Y	Y	Y	
	<color>	Y	Y	Y	
	transparent	Y	Y	Y	
	inherit	N	Y	Y	
background-image		Y	Y	Y	
	<uri>	Y	Y	Y	
	none	Y	Y	Y	
	inherit	N	Y	Y	
background-position		Y	Y	N	
	<percentage>{1,2}	Y	Y	N	
	<length>{1,2}	Y	Y	N	
	[top | center | bottom] || [left | center | right]	Y	Y	N	
	inherit	N	Y	N	
background-repeat		Y	Y	N	
	repeat	Y	Y	N	
	repeat-x	Y	Y	N	
	repeat-y	Y	Y	N	
	no-repeat	Y	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	inherit	N	Y	N	
color		Y	Y	Y	
	<color>	Y	Y	Y	
	inherit	N	Y	Y	

4.6 Fonts

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
font		Y	Y	Y	
	['font-style' || 'font-variant' || 'font-weight']?'font-size' [/ 'line-height']? 'font-family'	Y	Y	Y	
	caption	N	Y	N	
	icon	N	Y	N	
	menu	N	Y	N	
	message-box	N	Y	N	
	small-caption	N	Y	N	
	status-bar	N	Y	N	
	inherit	N	Y	Y	
font-family		Y	Y	N	
	[[<family-name> | <generic-family>],]* [<family-name> | <generic-family>]	Y	Y	N	
	inherit	N	Y	N	
font-size		Y	Y	N	
	<absolute-size>	Y	Y	N	
	<relative-size>	Y	Y	N	
	<length>	Y	Y	N	
	<percentage>	Y	Y	N	
	inherit	N	Y	N	
font-size-adjust		N	Y	N	
	number	N	Y	N	
	none	N	Y	N	
	inherit	N	Y	N	
font-stretch		N	Y	N	
	normal	N	Y	N	
	wider	N	Y	N	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	narrower	N	Y	N	
	ultra-condensed	N	Y	N	
	extra-condensed	N	Y	N	
	condensed	N	Y	N	
	semi-condensed	N	Y	N	
	semi-expanded	N	Y	N	
	expanded	N	Y	N	
	extra-expanded	N	Y	N	
	ultra-expanded	N	Y	N	
	inherit	N	Y	N	
font-style		Y	Y	Y	
	normal	Y	Y	Y	
	italic	Y	Y	Y	
	oblique	Y	Y	N	
	inherit	N	Y	Y	
font-variant		Y	Y	N	
	normal	Y	Y	N	
	small-caps	Y	Y	N	
	inherit	N	Y	N	
font-weight		Y	Y	Y	
	normal	Y	Y	Y	
	bold	Y	Y	Y	
	bolder	Y	Y	Y	
	lighter	Y	Y	Y	
	100	Y	Y	Y	
	200	Y	Y	Y	
	300	Y	Y	Y	
	400	Y	Y	Y	
	500	Y	Y	Y	
	600	Y	Y	Y	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	700	Y	Y	Y	
	800	Y	Y	Y	
	900	Y	Y	Y	
	inherit	N	Y	Y	
line-height		Y	Y	N	
	normal	Y	Y	N	
	number	Y	Y	N	
	<length>	Y	Y	N	
	<percentage>	Y	Y	N	
	inherit	N	Y	N	

4.7 Miscellaneous

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	
direction		N	Y	N	
display		Y	Y	Y	
	inline	Y	Y	N	
	block	Y	Y	N	
	list-item	Y	Y	N	
	run-in	N	Y	N	
	compact	N	Y	N	
	marker	N	Y	N	
	table	N	Y	N	
	inline-table	N	Y	N	
	table-row-group	N	Y	N	
	table-header-group	N	Y	N	
	table-footer-group	N	Y	N	
	table-row	N	Y	N	
	table-column-group	N	Y	N	
	table-column	N	Y	N	
	table-cell	N	Y	N	
	table-caption	N	Y	N	
	none	Y	Y	Y	
	inherit	N	Y	Y	
list-style		Y	Y	Y	
	'list-style-type' || 'list-style-position' || 'list-style-image'	Y	Y	Y	
	inherit	N	Y	Y	
list-style-image		Y	Y	N	
list-style-position		Y	Y	N	
list-style-type		Y	Y	Y	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	
	disc	Y	Y	Y	
	circle	Y	Y	N	
	square	Y	Y	N	
	decimal	Y	Y	Y	
	decimal-leading-zero	N	Y	N	
	lower-roman	Y	Y	N	
	upper-roman	Y	Y	N	
	lower-greek	N	Y	N	
	lower-alpha	Y	Y	Y	
	lower-latin	N	Y	Y	1
	upper-alpha	Y	Y	Y	
	upper-latin	N	Y	Y	1
	hebrew	N	Y	N	
	armenian	N	Y	N	
	georgian	N	Y	N	
	cjk-ideographic	N	Y	N	
	hiragana	N	Y	N	
	katakana	N	Y	N	
	hiragana-iroha	N	Y	N	
	katakana-iroha	N	Y	N	
	none	Y	Y	Y	
	inherit	N	Y	Y	
marker-offset		N	Y	N	
unicode-bidi		N	Y	N	
white-space		Y	Y	N	

4.8 Page

These properties are not supported

4.9 Selector Patterns

Pattern	CSS1	CSS2	Palm Web Pro 3.5	Notes
*	N	Y	Y	
E	Y	Y	Y	
E F	Y	Y	Y	
E > F	N	Y	N	
E:first-child	N	Y	N	
E:link	Y	Y	Y	
E:visited	Y	Y	Y	
E:active	Y	Y	N	
E:hover	N	Y	N	
E:focus	N	Y	N	
E:lang(c)	N	Y	N	
E + F	N	Y	N	
E[foo]	N	Y	N	
E[foo="warning"]	N	Y	N	
E[foo~="warning"]	N	Y	N	
E[lang|="en"]	N	Y	N	
DIV.warning	Y	Y	Y	
E#myid	Y	Y	Y	
E:first-letter	Y	Y	N	
E:first-line	Y	Y	N	
E:before	N	Y	N	
E:after	N	Y	N	
E, F	Y	Y	Y	

4.10 Tables

These properties are not supported

4.11 Text

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
letter-spacing		Y	Y	N	
	normal	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	
text-align		Y	Y	Y	
	left	Y	Y	Y	
	right	Y	Y	Y	
	center	Y	Y	Y	
	justify	Y	Y	N	
	<string>	N	Y	N	
	inherit	N	Y	Y	
text-decoration		Y	Y	Y	
	none	Y	Y	Y	
	underline	Y	Y	Y	
	overline	Y	Y	N	
	line-through	Y	Y	Y	
	blink	Y	Y	N	
	inherit	N	Y	Y	
text-indent		Y	Y	Y	1
	<length>	Y	Y	Y	1
	<percentage>	Y	Y	Y	1
	inherit	N	Y	Y	1
text-shadow		N	Y	N	
	none	N	Y	N	
	[<color> || <length> <length> <length>? ,]* [<color> || <length> <length> <length>?]	N	Y	N	
	inherit	N	Y	N	
text-transform		Y	Y	Y	

Properties	Values	CSS1	CSS2	Palm Web Pro 3.5	Notes
	capitalize	Y	Y	Y	
	uppercase	Y	Y	Y	
	lowercase	Y	Y	Y	
	none	Y	Y	Y	
	inherit	N	Y	Y	
vertical-align		Y	Y	N	
	baseline	Y	Y	N	
	sub	Y	Y	N	
	super	Y	Y	N	
	top	Y	Y	N	
	text-top	Y	Y	N	
	middle	Y	Y	N	
	bottom	Y	Y	N	
	text-bottom	Y	Y	N	
	<percentage>	Y	Y	N	
	<length>	N	Y	N	
	inherit	N	Y	N	
word-spacing		Y	Y	N	
	normal	Y	Y	N	
	<length>	Y	Y	N	
	inherit	N	Y	N	

4.12 User Interface

These properties are not supported

JavaScript Support Reference 5

The level of JavaScript support for Palm Web Pro will be roughly equivalent to JavaScript 1.5.

5.1 Invoking JavaScript

You can insert JavaScript in one of four ways:

- Event handlers:

```
<form name="frmInput" onsubmit="return MyFunction()">
```

- Inline script tags in an HTML page:

```
<script language="JavaScript">
var x;
function MyFunction(){
    x = frmInput.txtName.value;
    if (x == "Geoffrey"){
        return true;
        document.location.href = "http://mydomain.com/
redirect.asp";
    }
}
</script>
```

- Linking to script in an external file:

```
<script language="JavaScript" src="Header.js"></script>
```

- A “JavaScript URL:”

```
<a href="JavaScript:MyFunction();">Click here.</a>
```

5.2 Enable/Disable JavaScript

A user option will be provided to turn on or off JavaScript execution. The default is enabled.

5.3 The <NOSCRIPT> Tag

If JavaScript is not enabled, the HTML content specified by the **`<NOSCRIPT>`** tag on a page will be displayed. This allows developers to author pages which can display differently depending upon whether or not JavaScript support is enabled in the browser.

5.4 Error Handling

For Palm Web Pro, all JavaScript runtime errors encountered on a page will be ignored. Processing of additional **`<script>`** tags on the page will continue on in order to render as much of the page as possible.

5.5 Supported Elements

The following tables detail the JavaScript objects, properties, functions and events to be supported in Palm Web Pro:

Navigator		
Description	The Navigator object contains properties that describe the Web browser.	
Parent Object	None	
Properties	`appCodeName`	The code name of the browser. Default value is “Mozilla.”
	`appName`	The application name of the browser. Default value is “Netscape.”
	`appVersion`	The version number of the browser. Default value is 5.0 (windows; en-US).
	`cookieEnabled`	A Boolean which indicates whether the browser can store cookies.
	`mimeTypes`	A collection of supported MIME types. This array is always empty.
	`platform`	The operating system the browser is running under. Always contains “Win32.”
	`plugins`	An array of installed plugins. This collection is empty except for a single blank element. (This exists to make Macromedia’s Flash detection JavaScript code work properly.)
	`userAgent`	The HTTP user-agent value. Returns “<appCodeName>/<appVersion>.”
Methods	`javaEnabled()`	Indicates whether Java applets are supported. By default this method returns `false`. Palm Web Pro does not support Java applets, but configuring the Palm Web Pro Server to return `true` from this method is sometimes useful for some overly restrictive web sites.
Event Handlers	None	
JavaScript Syntax	`navigator` `window.navigator`	

window		
Summary	A Web browser window or frame.	
Description	The window object serves as the “global object,” and all expressions are evaluated in the context of the current browser window. The window object has `window` and `self` properties that refer to the window object itself. In addition to these, the parent and top properties and the `frames[]` array refer to other window objects related to the current one.	
Parent Object	None	
Properties	`defaultStatus`	The default status line text. This property has no effect since Palm Web Pro has no status bar.
	`document`	The current document in this window.
	`frames`	This collection contains a list of frames within the window.
	`location`	The URL of the window.
	`name`	Read-only string containing the name of the window. A window usually only has a meaningful name if it is a frame or iframe.
	`navigator`	Represents Palm Web Pro.
	`parent`	A reference to the containing window if this window is a frame or iframe.
	`self`	A reference to the window itself. A synonym for `window`.
	`status`	A read/write string that specifies the current contents of the status line. This property has no effect since Palm Web Pro has no status bar.
	`top`	A reference to the highest level window in the frameset.

window (Continued)		

Stubbed Properties	`innerHeight`
	`innerWIdth`
	`outerHeight`
	`outerWidth`
	`pageXOffset`
	`pageYOffset`
		These properties are recognized (that is, their usage does not throw a script error), but changing their values from your JavaScript code will have no noticeable result in Palm Web Pro.

window (Continued)		
Methods	`alert()`	Displays a message in a dialog box. The user must click OK to dismiss the dialog.
	`clearInterval()`	In proxyless mode, cancels further execution of the specified interval event, created via a previous setInterval() call. In proxy mode this method has no effect.
	`clearTimeout()`	In proxyless mode, cancels the scheduled execution of the specified timeout event, created via a previous setTimeout() call. In proxy mode this method has no effect.
	`confirm()`	Ask a yes-or-no question with a dialog box.
	`open()`	Open the URL specified in the first parameter. The second, third, and fourth parameters, which are optional, are ignored in Palm Web Pro.
	`prompt()`	Ask for simple string input with a dialog box.
	`setInterval()`	In proxyless mode, schedules repeated execution of the specified JavaScript code after the period specified. In proxy mode executes the specified JavaScript code once, immediately, if the interval specified is no more than 60 seconds.
	`setTimeout()`	In proxy mode runs the specified JavaScript code after the specified amount of time. In proxy mode executes the specified JavaScript code immediately if the interval specified is no more than 60 seconds.

window (Continued)		
Event Handler	`onLoad`	The handler invoked when a document finishes loading.

JavaScript Syntax	`self`
	`window`
	`window.frames[`*`frame_name`*`]`	

See Also	`document`	

document		
Description	The `document` object represents the HTML document displayed in a browser window or frame. The properties of this object provide details about many aspects of the document.The `write()` method of the document object is especially notable. When invoked in scripts that are run while the document is loading, you can call `document.write()` to insert dynamically generated HTML text into the document.	
Parent Object	`window`	
Properties	`all`	A collection of all frames, iframes, forms and input elements in the window. Objects in this collection can be accessed using an integer array index, or more commonly as a property of the array object, using the name assigned by the `NAME` or `ID` attribute of the HTML tag.
	`anchors`	A collection of anchor objects (`A` tags), one for each hypertext target in the document. This collection is always empty.
	`applets`	A collection of Java objects, one for each `APPLET` tag that appears in the document. This collection is always empty.
	`cookie`	A read-write string containing all cookies for the current page.
	`embeds`	An array of Java objects, one for each `EMBED` tag that appears in the document. This collection is always empty.
	`forms`	An array of form objects, one for each `FORM` tag that appears in the document.
	`images`	An array of image objects, one for each image embedded in the document with the `IMG` tag. This collection is always empty.

document (Continued)		
	`links`	An array of link objects, one for each hypertext link in the document. This collection is always empty.

	`location`
	`plugins`
	`title`
	`URL`
		A location object representing the current page’s URL. Implemented as a reference to window.location.
		A synonym for the `embeds` array. This collection is always empty.
		A read-only string that specifies the contents of the `TITLE` tag of the document.
		A read-only string that specifies the URL of the document.

Methods	`getElementByID()`
	`write()`
	`writeln()`
		Retrieves a reference to any element in a document that has a unique identifier assigned to its `ID` attribute.
		Inserts the specified string into the document currently being parsed.
		A method identical to `write()`, except that it appends a newline character to the output.

Event Handlers	None	
JavaScript Syntax	`window.document` `document`	
HTML Syntax	The document object obtains values for a number of its properties from attributes of the HTML `BODY` tag. Also, the HTML contents of a document appear between `<BODY>` and `</BODY>`.	
See Also	`form`, `window`	

form		
Description	A form object represents an HTML `FORM` tag in a document. Each form in a document is represented as an element of the collection `document.forms[].` To reference a form, specify its index number or its name (the value of the `NAME` attribute on the `FORM` tag), such as `document.forms.`***`frmLogOn`***.	
Parent Object	`document`	
Properties	`action`	A read/write string specifying the URL to which the form will be submitted.
	`children`	Not supported on the Palm Web Pro Server.
	`elements`	A collection of input elements that appear on the form. Each element is a button, check box, hidden, password, radio, reset, select, submit, text, or text area object.
	`elements.length`	The number of items in the `elements` array.
	`encoding`	A read/write string that specifies the encoding method used for form data.
	`length`	The number of elements in the form. Equivalent to `elements.length`.
	`method`	A read/write string that specifies the technique for submitting the form. It should have the value "get" or "post."
	`name`	The name of the form.
	`referrer`	When a link from one document leads to another, the second document can, under JavaScript control, reveal the URL of the document containing the link.
	`target`	Not supported on the Palm Web Pro Server.

form (Continued)		
Methods	`submit()`	Submits the form to the URL specified in the **`ACTION`** attribute of the **`FORM`** tag.
	`reset()`	This method is recognized (that is, its usage does not throw a script error), but it has no effect in Palm Web Pro.
Event Handlers	`onSubmit`	Invoked just before the form is submitted.

JavaScript Syntax	`document.`*`form_name`*
	`document.forms[`*`form_number`*`]`	

HTML Syntax	A form object is created with a standard HTML **`FORM`** tag.	
See Also	`Input Elements`	

Input Elements		
Description	Form elements are stored in the **`elements[]`** array of the form object. The contents of this array are input objects, which represent the individual buttons, input fields, and other controls that appear within the form. The various form input elements share quite a few properties, methods, and event handlers. Behavior that is specific to a type of form element is indicated below.	
Parent Object	**`form`**	
Properties	**`checked`**	A read/write Boolean that indicates the selection state of form element. Applies to a check box and radio button.
	`defaultChecked`	A read-only Boolean that specifies the default state of a form element that can be checked. Applies to check box and radio button.
	`defaultValue`	A read-only string that specifies the default value for a form element. Applies to text, text area, and password.
	`form`	A read-only reference to the form object that contains this element.
	`id`	A read-only string that specifies the value of the id attribute of this element.
	`length`	For the select form element (a drop-down list created with the **`SELECT`** tag), this property specifies the number of options or choices (each represented by an Option object) that are contained in the **`options`** collection.
	`name`	A read-only string that specifies the name of this element. This name may be used to refer to the element, as shown in the JavaScript syntax below.

Input Elements (Continued)		
	`options`	For the SELECT form element, this collection contains option objects that represent the choices displayed in the select object. See “Option,” below.
	`selectedIndex`	For the select form element, this integer specifies which of the options displayed by the `SELECT` object is currently selected.
	`type`	A read-only string that specifies the type of the `form` element.
	`value`	A string property that specifies the value to be sent to the server for this element when the form that contains it is submitted.
Methods	`add()`	Belongs to the options array property of a `SELECT` element object. Inserts the specified Option object to the specified position in the `options` array, shifting any existing elements down one in the options array.
	`blur()`	Generates a blur event. This method is stubbed out on the Palm Web Pro Server only.
	`focus()`	Generates a focus event. This method is stubbed out on the Palm Web Pro Server only.
	`click()`	Allows JavaScript code to perform the same action on an input element as a user’s clicking on that element would.
	`remove()`	If the element is a SELECT, this method can be called to remove an OPTION from the SELECT.
Event Handlers	`onBlur`	Executes the specified JavaScript code or function on the occurrence of a blur event. This event handler is not supported on the Palm Web Pro Server.

Input Elements (Continued)		
	`onClick`	For form elements that are buttons, this event handler is invoked when the user clicks or selects the button.
	`onFocus`	Executes the specified JavaScript code or function on the occurrence of a focus event. This event handler is not supported on the Palm Web Pro Server.
	`onChange`	Executes the specified JavaScript code or function on the occurrence of a change event. On the Palm Web Pro Server, this event handler is supported only for the SELECT form elements.
JavaScript Syntax	***form*.elements[*i*]** ***form.input_name***	
See Also	`Option`	

Option		

Description	This object represents a single option displayed within a select object. A select object is created with the HTML tag `SELECT`. The following script fragment demonstrates one way to retrieve an option object from the select tag named selStates on the form named frmUserLoc:
	`var oForm, oSelect, oOption;`
	`oForm = document.forms.frmUserLoc;`
	`oSelect = oForm.elements.selStates;`
	`oOption = oSelect.options[2];`	

Parent Object	An `input` element of type `select`	
Properties	`defaultSelected`	A read-only Boolean that specifies whether this option is selected by default.
	`index`	A read-only integer that specifies the index of this option within the `options` collection.
	`selected`	A read/write Boolean that specifies whether this option is currently selected.
	`text`	The text that describes the option. It is the plain text (not formatted HTML text) that follows the `OPTION` tag.
	`value`	A read/write string that specifies the value to be passed to the server if this option is selected when the form is submitted.
Methods	`Option (text, value [, default selected, selected])`	Create a new `Option` that can then be added to an existing `SELECT.`
Event Handlers	None	
JavaScript Syntax	`*select*.options[*i*]`	
See Also	`Input Elements`	

location		
Description	The location object represents the Web address of the document currently displayed in the window.The properties of a location object refer to the various portions of a URL, which has the following format: `protocol://hostname:port/pathname?search#hash`	
Parent Object	`window`	
Properties	`hash`	The hash portion of the URL, including the leading hash (#) mark. Note that the Adaptive Content Accelerator removes `<A NAME="">` tags, as well as the hash in `HREF` attributes; therefore, you can retrieve the hash value only if it was typed directly into the location field.
	`host`	A combination of the hostname and port portions of the URL.
	`hostname`	The hostname portion of the URL
	`href`	The complete URL.
	`pathname`	The path portion of the URL.
	`port`	The port portion of the URL.
	`protocol`	The protocol portion of the URL.
	`search`	The search, or query, portion of the URL, including the leading question mark.
Methods	`reload()`	This method is recognized (that is, its usage does not throw a script error), but it has no effect in Palm Web Pro.
	`replace()`	Replace the current document with a new one.
Event Handlers	None	
JavaScript Syntax	`location` `window.location`	
See Also	`Input Elements`	

Image		
Description	This object is an image on an HTML form, created by using the HTML `IMG` tag. Any images created in a document are then stored in an array in the `document.images` property, and it is from here that they are accessed. To define an image with its constructor, use the following syntax: `new Image(width, height)`	
Parent Object	`document`	
Properties	`src`	A string representing the URL of an image and reflects the SRC attribute of an `IMG` tag.
Methods		
Event Handlers	None	
JavaScript Syntax	`document.images`	
See Also		

Pluggable Scheme Handlers 6

6.1 Overview

A Pluggable Scheme Handler (PSH) is a software extension to the browser that can be created by third party developers. A PSH gives the ability to extend the browser to handle schemes that are not natively supported. For example the FTP (File Transfer Protocol) is not natively supported in the browser but could be added with a PSH. The scheme is the first part of a URL defined in RFC 2396 identified as "scheme". For example HTTP: is the scheme that indicates to a browser that the HTTP protocol should be used to retrieve the content addressed by the rest of the URL.

6.2 Basics

6.2.1 Transaction Lifecycle

The term transaction here refers to the steps needed, given a URL, to route a request to the correct scheme handler and end up with content addressed by the URL.

When the browser is required to retrieve a URL for any reason these steps will be taken.

1. The http cache will be queried for a match using standard http cache policies. This will occur regardless of the scheme type, internal or PSH. If a match is found we're done.
2. Determine the scheme name (http, ftp, etc.).
3. If the scheme can be handled internally in the browser it will do so without talking to a PSH. In this case it's handled internally and outside of the scope of this document, therefore ignore the remaining steps.
4. If a PSH exists that supports the scheme, continue, otherwise the browser is not able to retrieve the content, therefore ignore the remaining steps.

5. If the PSH shared library for this scheme is not currently loaded, it will be loaded now.
6. The browser will call the `PshLibOpen()` method. The PSH can create a context at this point.
7. The browser will call the `PshLibStart()` method passing all information about the request. At this point the PSH may decide to forward this request to the appropriate host.
8. The browser will call the `PshLibDescribe()` method. This will be called repeatedly until the `PshLibErrDataComplete` response code is received. During each call to the `PshLibDescribe()` method the PSH can return a single piece of information that describes the content. This must take the form of an HTTP header. For example: Content-Type: text/html.
9. The browser will call the `PshLibRead()` method. This will be called repeatedly until the `PshLibErrDataComplete` response code is received. During each call to `PshLibRead()` the PSH can return content to the browser.
10. The browser will call the `PshLibTerminate()` method.
11. The browser will call the `PshLibClose()` method.
12. If there are no other active transactions open to this PSH it will be unloaded from memory.

6.2.2 Creating a PSH

A PSH is a shared library that implements a defined set of APIs. The easiest way to create a PSH is to modify an existing sample PSH. However, a new PSH can be created from scratch or created with a shared library wizard that may be included in the development environment.

A PSH must have a type ID of '`psh `' (that is '`psh`' with a space). Any creator ID is valid.

The PshLib.h header must be included in the shared library project. This defines the entry points into the shared library. This PshLib.h file should not be modified.

6.2.3 Required PSH Implementation

A PSH must implement a minimum set of API in order to function properly with the browser.

- The `PshLibInfo()` method must be implemented and return a scheme name.
- The `PshLibDescribe()` must return the `PshLibErrDataComplete` response code when it has no more description data to send to the browser.

- The `PshLibRead()` must return the `PshLibErrDataComplete` response code when it has no more data to send to the browser.

6.2.4 PSH Global Data

A PSH, as any shared library, can contain global data. This is outside of the requirements of the PSH system and completely independent of the browser. The `PshHttpSample` sample has an example of global data usage.

Be aware that the browser will load and unload the PSH as it needs to, so don't depend on the PSH being loaded into memory throughout the lifetime of the browser.

6.2.5 PSH Context Data

Each call to a PSH that comes from the browser includes a 4-byte context value that the PSH can use to reference instance specific information. The value is unused by the browser, it's only purpose is to allow the PSH to map back to the correct context. The PSH can only create this context during the `PshLibOpen()` call. Note that the Wake and Sleep method do not come from the browser and therefore do not include the context.

A PSH must be written with the knowledge that browser may open multiple instances to a single PSH. For this reason most PSH's will need context data. Currently the browser will open at most 4 instances of a PSH, but a PSH implementation should not depend on this current maximum, if may increase or decrease.

The `PshHttpSample` sample has an example of context data usage.

6.2.6 Using NetLib from a PSH

A PSH can use the NetLib for networking. It will need to manage the opening of the library, sockets and connections on it's own. The PSH should plan on the common case that NetLib is also concurrently in use by the browser. The PSH must free network resources when it is no longer using them.

The `PshHttpSample` sample is an example of using the NetLib networking library from a PSH.

6.2.7 Describing Content

A PSH is not required to describe the content is deliveries to the browser. However, there is a benefit to do so. The browser will treat the content from a PSH just like all content it receives itself from natively handled schemes.

The `PshLibDescribe()` method is the only opportunity for the PSH to give this content description information to the browser. The form this description takes is very closely mapped to HTTP headers.

The Content-Type header is one that a PSH developer should seriously consider returning to the browser. While the browser does try to sniff the content to determine the content type, if it can't determine the content based on content examination it will then rely on the Content-Type header.

Another header or series of headers a PSH developer should consider are the cache related headers. If no cache control or expiry headers are found the browser will use default behavior. These give the PSH the opportunity to control the caching of content it gives to the browser.

6.2.8 Accessing Resources from a PSH

Resources can be accessed from a PSH as one would any shared library. To aid in this process the browser will open the PSH database prior to calls into the PSH. This will ensure that the PSH database is on the top of the database stack that is used for resource searching. If the Palm API **`DmGetResouce()`** is used instead of **`DmGet1Resource()`** there is a chance that unexpected resources could be retrieved, if a PSH database doesn't have the requested resource.

6.2.9 User Interface from a PSH

Building on the information by the section Accessing Resources from a PSH. A PSH can display user interfaces but should do so with caution. The browser and other PSH's may also attempt to display a user interface as well. The browser will display a number of forms as pop-ups, and are non-blocking. So care should be taken if a PSH also wishes to pop-up a modeless form that it's form is on top before writing to it.

In addition the use of a modal dialog will stop all functionality in the browser until the form is closed.

The form resource range 9000 - 9999 is reserved for browser plug-ins. This includes PSH's and any future plug-in system.

The PshLib2 sample is an example of displaying a form in a PSH.

6.2.10 Installing a PSH

A PSH need only be placed on the device to be installed. The browser will search for all PSH's on start-up. In addition if the browser is used to download a PSH it will also locate and be come aware of the PSH without requiring a restart of the browser.

6.3 Samples

6.3.1 PshLib (Skeleton)

This is a skeleton PSH that returns a fixed content message to the browser.

The scheme name used in the sample is `pshdev`.

For example a URL of `pshdev://test.me.com` will use this PSH.

6.3.2 PshLib2 (User Interface)

This is a sample PSH that displays an empty form as an example of UI in a PSH. In addition it also shows usage of global and context data.

The scheme name used in the sample is pshdev2.

For example a URL of pshdev2://test.me.com will use this PSH.

6.3.3 PshHttpSample (Very simple HTTP in a PSH)

This is a sample PSH that implements a very minimum HTTP protocol. It should be stressed that this is a very incomplete HTTP implementation and intended only to show a sample of doing networking from a PSH.

The scheme name used in this sample is pshhttpsample

For example a URL of `pshhttpsample://www.google.com` will retrieve the Google page.

6.4 Reference

6.4.1 API

6.4.1.1 PshLibOpen

PshLibOpen		
Purpose	Opens the library, creates and initializes the globals. This function must be called before any other library functions.	
Prototype	Err PshLibInfo(UInt16 refNum, UInt32 *clientContextP)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	Pointer to variable for returning client context. The client context is used to maintain client-specific data for multiple client support. The value returned here will be used as a parameter for other library method that require a client context.
Results	ErrNone or memErrNotEnoughSpace may be retuned from this method.	
Comments	This is one of the standard 4 shared library entry points.	

6.4.1.2 PshLibClose

PshLibClose		
Purpose	Closes the library, frees client context and globals.	
Prototype	Err PshLibInfo(UInt16 refNum, UInt32 *clientContextP)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	The context.
Results	ErrNone is the only expected result.	
Comments	This is one of the standard 4 shared library entry points.	

6.4.1.3 PshLibInfo

PshLibInfo		
Purpose	Returns scheme name to the browser.	
Prototype	Err PshLibInfo(UInt16 refNum, char **scheme, UInt16 *version)	
Parameters	-> refNum	The library reference number.
	<- scheme	Pass an address of a pointer, which the PSH must set to point to a character string of the scheme name.
	<- version	Pass an address of an integer that the PSH must set to the version of the PSH API implemented.
Results	ErrNone is the only expected result.	
Comments	A context in not passed to this method since it is intended to be somewhat static.	

6.4.1.4 PshLibStart

PshLibStart		
Purpose	Pass request information to a PSH.	
Prototype	Err PshLibStart(UInt16 refNum, UInt32 clientContextP, const char *uri, const char *method, const char *host, UInt16 port, const char *path, const char *userAgent, const char *accept, const char *acceptCharset, const char *acceptLanguage, const char *acceptEncoding, const char *data)	
Parameters	-> refNum	The library reference number.
	-> clientContextP	The context.
	-> uri	The full URL including scheme. For example pshdev://test.me.com
	-> method	The method that the browser would have used if this was an HTTP request. Can be "GET" or "POST"
	-> host	A pointer to the host part of the URL. This will be with out any additional delimiters. For example in the case of the URL http://www.google.com the host string will be "www.google.com".
	-> port	The host port from the URL
	-> path	The path and query string part of the URL.
	-> userAgent	The browser user agent string. This corresponds to the HTTP header User-Agent:

PshLibStart		
	-> accept	The content types the browser can handle. This corresponds to the HTTP header Accept:
	-> acceptCharset	The character sets the browser can handle. This corresponds to the HTTP header Accept-Charset:
	-> acceptLanguage	The language the browser can handle. This corresponds to the HTTP header Accept-Language:
	-> acceptEncoding	The encoding types the browser can handle. This corresponds to the HTTP header Accept-Encoding:
	-> data	This may include the equivalent to posted data in the HTTP protocol.
Results	Any of the PSH error codes can be returned here.	
Comments		

6.4.1.5 PshLibTerminate

PshLibTerminate		
Purpose	Informs the PSH that it should terminate the transaction started with PshLibStart()	
Prototype	Err PshLibTerminate(UInt16 refNum, UInt32 clientContextP)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	The context.
Results	ErrNone is the only expected result.	
Comments		

6.4.1.6 PshLibRead

PshLibRead		
Purpose	Returns data to the browser	
Prototype	Err PshLibRead(UInt16 refNum, UInt32 clientContextP, UInt8 *buffer, UInt32 bufferSize, UInt32 *bytesWritten)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	The context.
	<- buffer	Pointer to a buffer that can be filled with data.
	-> bufferSize	The size of the buffer. This is the amount of data that be can placed in the buffer.
	<- bytesWritten	A pointer to an integer that the PSH must set to the number of bytes placed in buffer.
Results	PshLibErrWouldBlock	The PSH must return this result when it has no has data to return but wants to be called again to send more data to the browser.
	errNone	The PSH returned some data.
	PshLibErrDataComplete	The PSH has no more data to send to the browser.
Comments		

6.4.1.7 PshLibDescribe

PshLibDescribe		
Purpose	Returns content description data to the browser	
Prototype	Err PshLibDescribe(UInt16 refNum, UInt32 clientContextP, Int8 *buffer, UInt32 bufferSize, UInt32 *bytesWritten)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	The context.
	<- buffer	Pointer to a buffer that can be filled with data.
	-> bufferSize	The size of the buffer. This is the amount of data that be can placed in the buffer.
	<- bytesWritten	A pointer to an integer that the PSH must set to the number of bytes placed in buffer.
Results	PshLibErrWouldBlock	The PSH must return this result when it has no has data to return but wants to be called again to send more data to the browser.
	errNone	The PSH returned some data.
	PshLibErrDataComplete	The PSH has no more data to send to the browser.
Comments		

6.4.1.8 PshLibErrorMessage

PshLibErrorMessage		
Purpose	Returns an error message string to the browser	
Prototype	Err PshLibRead(UInt16 refNum, UInt32 clientContextP, Int8 *buffer, UInt32 bufferSize, UInt32 *bytesWritten)	
Parameters	-> refNum	The library reference number.
	<- clientContextP	The context.
	<- buffer	Pointer to a buffer that can be filled with data.
	-> bufferSize	The size of the buffer. This is the amount of data that be can placed in the buffer.
	<- bytesWritten	A pointer to an integer that the PSH must set to the number of bytes placed in buffer.
Results	errNone	The PSH returned some data.
Comments		

Integrating Applications with Palm Web Pro 7

The following sections describe a few ways in which your applications can take advantage of certain features of Palm Web Pro 3.5:

- "Launching an Application from a Web Page" on page 7-1
- "Opening a Web Page from an Application" on page 7-2
- "Registering Applications with Exchange Manager" on page 7-4

Note:

Some of these techniques involve launching or sublaunching an application. For a thorough discussion of the difference between launching and sublaunching, see the Palm OS SDK documentation.

7.1 Launching an Application from a Web Page

You can use two different URL schemes to launch a Palm OS application from your web page.

- Use the Palm: scheme to launch an application.
- Use the PalmCall: scheme to sublaunch an application. Remember that a launched application can access globals, but a sublaunched application cannot.

The syntax for launching or sublaunching an application from a web page is very similar:

```
<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 or sublaunch, and textstring is the text displayed in Palm Web Pro.

7.1.1 Passing Data Back and Forth

You can pass data from your web page 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. Separate the arguments by an ampersand (&). For example:

```
<a href="palm:MYAP.appl?name=PalmOne, Inc.&city=Milpitas">Call My
App</a>
```

For an example that shows how an application would handle both the Palm: and PalmCall: cases, see the Bars sample application in the Palm OS SDK Web Browser Update.

7.2 Opening a Web Page from an Application

The Palm OS® provides two ways to open web pages from an application using Palm Web Pro. You can launch Palm Web Pro from another Palm OS application through the Palm Exchange Manager or you can launch or sublaunch Palm Web Pro directly from another application.

7.2.1 Using Exchange Manager

Palm Web Pro is registered with the Exchange Manager for the http and https schemes and will respond to requests to handle URLs.

The example below shows how to write a function that will launch Palm Web Pro to handle a URL using the Exchange Manager.

```
/***********************************************************************/
Err SendURL(char *address)
{
  Err err = errNone;
  ExgSocketType exgSocket;

  // init to zero
  ::MemSet(&exgSocket, sizeof(exgSocket), 0);

  exgSocket.name = (char *) ::MemPtrNew(::StrLen(address)+1);
  ::StrCopy(exgSocket.name, (const char *)address);
  exgSocket.length = ::StrLen(address);
  exgSocket.count = 1;
  // init localMode to indicate "send data back to the same device"
  exgSocket.localMode = 1;

  err = ::ExgRequest(&exgSocket);
  ::MemPtrFree(exgSocket.name);

  if (err != errNone)
  {
    char buffer[100];
    ::StrPrintF(buffer, "Error %d: Failed to exchange URL.", err);
    // pop an alert
    ::FrmCustomAlert(CustomInformationAlert, "", buffer, "");
  }
  return err;
}
```

7.2.2 Launching Palm Web Pro directly

You can launch Palm Web Pro from a Palm OS application by using the **`sysAppLaunchCmdGoToURL`** launch code.

The example below shows how to write a function that will launch Palm Web Pro to its Web Content dialog with the appropriate arguments.

```
#include <PalmOS.h>
Err WebBrowserCommand(UInt16 launchFlags, UInt16 command, Char *parameterP,
UInt32 *resultP)
{
  UInt16             cardNo;
  LocalID            dbID;
  DmSearchStateType  searchState;
  Err                error;
  if (resultP) *resultP = errNone;

  error = DmGetNextDatabaseByTypeCreator(true, &searchState,
  sysFileTApplication, ‘NOVR’, true, &cardNo, &dbID);

  if (error == errNone)
  {
    {
      Char *newParamP = NULL;
      if (parameterP)
      {
        newParamP= MemPtrNew( StrLen(parameterP) +1 );
        if (newParamP == NULL) error = memErrNotEnoughSpace;
        else
        {
          ::StrCopy(newParamP, parameterP);
          ::MemPtrSetOwner(newParamP, 0); // The OS now owns this memory
        }
      }
      if (error == errNone)
      {
        SysUIAppSwitch( cardNo, dbID, command, newParamP);
      }
    }
  }
  return error;
} //WebBrowserCommand
```

To open a specific web page by launching Palm Web Pro from your application, call this function using a command similar to the following example:

```
WebBrowserCommand(0, sysAppLaunchCmdGoToURL, "www.palm.com", NULL);
```

Note that the system ignores the `launchFlags` and `resultP` parameters in this example.

7.3 Registering Applications with Exchange Manager

The Palm OS Exchange Manager lets you tell the system which applications should handle which types of content. For more information about Exchange Manager, see the Palm OS SDK documentation.

7.3.1 Handling Web Content

Palm Web Pro itself is registered to handle the following schemes: http and https. It is registered with Exchange Manager for handling these types.

7.3.2 Handling Other Content Types

Palm Web Pro is able to download files from the Internet/intranet. If an application on the device supports the downloaded file type, Palm Web Pro will send the file to the application via the Exchange Manager.

You can register an application to handle any content type by calling Exchange Manager from within the application. An application can register for a particular file extension, mime type, creator ID, or URL scheme. For instance, the example below shows how to register a mail application to handle the `mailto:` scheme.

```
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, Exchange Manager makes that application the default handler for email messages. This means that the newly registered application becomes the application that the Exchange Manager calls when Palm Web Pro encounters the `mailto:` scheme.

Palm Web Pro also handles the `phoneto:` scheme through the Exchange Manager, calling the registered application to handle the phone number.

Supported Certificates A

A.1 Proxyless Supported Certificates

Autoridad Certificadora de la Asociacion Nacional del Notariado .cer

Autoridad Certificadora del Colegio Nacional de Correduria Publica Mexicana, A.C..cer

Belgacom E-Trust Primary CA.cer

Certiposte Editeur.cer

Certiposte Serveur.cer

Certisign Autoridade Certificadora AC1S.cer

Certisign Autoridade Certificadora AC2.cer

Certisign Autoridade Certificadora AC3S.cer

Certisign Autoridade Certificadora AC4.cer

CertPlus Class 1 Primary CA.cer

CertPlus Class 2 Primary CA.cer

CertPlus Class 3 Primary CA.cer

CertPlus Class 3P Primary CA.cer

CertPlus Class 3TS Primary CA.cer

Deutsche Telekom Root CA 1.cer

Deutsche Telekom Root CA 2.cer

dsk-conslt.cer

DST (ABA.ECOM) CA.cer

DST (ANX Network) CA.cer

DST (Baltimore EZ) CA.cer

DST (National Retail Federation) RootCA.cer

DST (United Parcel Service) RootCA.cer

DST RootCA X1.cer

DST RootCA X2.cer

DST-Entrust GTI CA.cer

DSTCA E1.cer

DSTCA E2.cer

Entrust.net Secure Server Certification Authority.cer

Equifax Secure Certificate Authority.cer

Equifax Secure eBusiness CA-1.cer

Equifax Secure eBusiness CA-2.cer

Equifax Secure Global eBusiness CA-1.cer

EUnet International Root CA.cer

Fabrica Nacional de Moneda y Timbre.cer

FESTE, Public Notary Certs.cer

FESTE, Verified Certs.cer

First Data Digital Certificates Inc. Certification Authority.cer

GlobalSign Root CA.cer

GTE CyberTrust Global Root.cer

GTE CyberTrust Root (1).cer

GTE CyberTrust Root.cer

IPS SERVIDORES.cer

Japan Certification Services, Inc. SecureSign RootCA1.cer

Japan Certification Services, Inc. SecureSign RootCA2.cer

Japan Certification Services, Inc. SecureSign RootCA3.cer

KeyMail PTT Post Root CA.cer

Microsoft Root Authority.cer

NetLock Expressz (Class C) Tanusitvanykiado.cer

NetLock Kozjegyzoi (Class A) Tanusitvanykiado.cer

NetLock Uzleti (Class B) Tanusitvanykiado.cer

Saunalahden Serveri CA .cer

Saunalahden Serveri CA.cer

SERVICIOS DE CERTIFICACION - A.N.C..cer

Societa Interbancaria per l'Automazione SIA Secure Client CA.cer

Societa Interbancaria per l'Automazione SIA Secure Server CA.cer

Swisskey Root CA.cer

TC TrustCenter Class 1 CA.cer

TC TrustCenter Class 2 CA.cer

TC TrustCenter Class 3 CA.cer

TC TrustCenter Class 4 CA.cer

TC TrustCenter Time Stamping CA.cer

Thawte Personal Basic CA.cer

Thawte Personal Freemail CA.cer

Thawte Personal Premium CA.cer

Thawte Premium Server CA.cer

Thawte Server CA.cer

Thawte Timestamping CA.cer

UTN - DATACorp SGC.cer

UTN - USERFirst-Client Authentication and Email.cer

UTN - USERFirst-Hardware.cer

UTN - USERFirst-Network Applications.cer

UTN - USERFirst-Object.cer

ValiCert Class 1 Policy Validation Authority.cer

ValiCert Class 2 Policy Validation Authority.cer

ValiCert Class 3 Policy Validation Authority.cer

VeriSign Class 1 Primary CA (10).cer

VeriSign Class 1 Primary CA (8).cer

VeriSign Class 1 Primary CA.cer

VeriSign Class 1 Public Primary CA.cer

VeriSign Class 2 Primary CA (4).cer

VeriSign Class 2 Primary CA (5).cer

VeriSign Class 2 Primary CA.cer

VeriSign Class 2 Public Primary CA.cer

VeriSign Class 3 Primary CA (6).cer

VeriSign Class 3 Primary CA (7).cer

VeriSign Class 3 Primary CA.cer

VeriSign Class 3 Public Primary CA.cer

VeriSign Class 4 Primary CA (9).cer

VeriSign Class 4 Primary CA.cer

VeriSign Commercial Software Publishers CA (3).cer

VeriSign Individual Software Publishers CA (3).cer

VeriSign Time Stamping CA.cer

verisign_01.cer

VeriSign_RSA Secure Server CA.cer

ViaCode Certification Authority.cer

Xcert EZ by DST.cer

A.2 Palm Web Pro Server Supported Certificates

Autoridad Certificadora de la Asociacion Nacional del Notariado .cer

Autoridad Certificadora del Colegio Nacional de Correduria Publica Mexicana, A.C..cer

Belgacom E-Trust Primary CA.cer

Certiposte Editeur.cer

Certiposte Serveur.cer

Certisign Autoridade Certificadora AC1S.cer

Certisign Autoridade Certificadora AC2.cer

Certisign Autoridade Certificadora AC3S.cer

Certisign Autoridade Certificadora AC4.cer

CertPlus Class 1 Primary CA.cer

CertPlus Class 2 Primary CA.cer

CertPlus Class 3 Primary CA.cer

CertPlus Class 3P Primary CA.cer

CertPlus Class 3TS Primary CA.cer

Deutsche Telekom Root CA 1.cer

Deutsche Telekom Root CA 2.cer

DST (ABA.ECOM) CA.cer

DST (ANX Network) CA.cer

DST (Baltimore EZ) CA.cer

DST (National Retail Federation) RootCA.cer

DST (United Parcel Service) RootCA.cer

DST RootCA X1.cer

DST RootCA X2.cer

DST-Entrust GTI CA.cer

DSTCA E1.cer

DSTCA E2.cer

Entrust.net Secure Server Certification Authority.cer

Equifax Secure Certificate Authority.cer

Equifax Secure eBusiness CA-1.cer

Equifax Secure eBusiness CA-2.cer

Equifax Secure Global eBusiness CA-1.cer

EUnet International Root CA.cer

Fabrica Nacional de Moneda y Timbre.cer

FESTE, Public Notary Certs.cer

FESTE, Verified Certs.cer

First Data Digital Certificates Inc. Certification Authority.cer

GlobalSign Root CA.cer

CyberTrust Global Root.cer

GTE CyberTrust Root (1).cer

GTE CyberTrust Root.cer

IPS SERVIDORES.cer

Japan Certification Services, Inc. SecureSign RootCA1.cer

Japan Certification Services, Inc. SecureSign RootCA2.cer

Japan Certification Services, Inc. SecureSign RootCA3.cer

KeyMail PTT Post Root CA.cer

Microsoft Root Authority.cer

NetLock Expressz (Class C) Tanusitvanykiado.cer

NetLock Kozjegyzoi (Class A) Tanusitvanykiado.cer

NetLock Uzleti (Class B) Tanusitvanykiado.cer

Saunalahden Serveri CA .cer

Saunalahden Serveri CA.cer

SERVICIOS DE CERTIFICACION - A.N.C..cer

Societa Interbancaria per l'Automazione SIA Secure Client CA.cer

Societa Interbancaria per l'Automazione SIA Secure Server CA.cer

Swisskey Root CA.cer

TC TrustCenter Class 1 CA.cer

TC TrustCenter Class 2 CA.cer

TC TrustCenter Class 3 CA.cer

TC TrustCenter Class 4 CA.cer

TC TrustCenter Time Stamping CA.cer

Thawte Personal Basic CA.cer

Thawte Personal Freemail CA.cer

Thawte Personal Premium CA.cer

Thawte Premium Server CA.cer

Thawte Server CA.cer

Thawte Timestamping CA.cer

UTN - DATACorp SGC.cer

UTN - USERFirst-Client Authentication and Email.cer

UTN - USERFirst-Hardware.cer

UTN - USERFirst-Network Applications.cer

UTN - USERFirst-Object.cer

ValiCert Class 1 Policy Validation Authority.cer

ValiCert Class 2 Policy Validation Authority.cer

ValiCert Class 3 Policy Validation Authority.cer

VeriSign Class 1 Primary CA (10).cer

VeriSign Class 1 Primary CA (8).cer

VeriSign Class 1 Primary CA.cer

VeriSign Class 1 Public Primary CA.cer

VeriSign Class 2 Primary CA (4).cer

VeriSign Class 2 Primary CA (5).cer

veriSign Class 2 Primary CA.cer

VeriSign Class 2 Public Primary CA.cer

VeriSign Class 3 Primary CA (6).cer

VeriSign Class 3 Primary CA (7).cer

VeriSign Class 3 Primary CA.cer

VeriSign Class 3 Public Primary CA.cer

VeriSign Class 4 Primary CA (9).cer

VeriSign Class 4 Primary CA.cer

VeriSign Commercial Software Publishers CA (3).cer

VeriSign Individual Software Publishers CA (3).cer

VeriSign Time Stamping CA.cer

veriSign_RSA Secure Server CA.cer

ViaCode Certification Authority.cer

www.verisign.comCPS Incorp.by Ref.LIABILITY LTD.(c)97 VeriSign.cer

Xcert EZ by DST.cer