

Getting Started with WebC

EBS Embedded Software

Revised 2/1/13

Contents

Getting Started with WebC.....	1
Feature Overview.....	1
Documentation Roadmap.....	2
WebC Architecture.....	2
Porting WebC.....	4
Simple Example Application.....	5
Building WebC.....	5
Configuring WebC.....	6
Compile-time Configuration	6
Run-time Configuration	6
Appendix A.....	6
Appendix B.....	10
Appendix C.....	17

Feature Overview

WebC is a highly capable SDK that offers tremendous power and flexibility to the graphical embedded application developer. WebC is compatible with HTML version 4.01, Cascading Style Sheets level 1 (with support for much of CSS/2), JavaScript 1.5, HTTP 1.1 (with support for Basic authentication, persistent connections, and memory or disk-based caching), GIF images and animations, JPEG images, PNG images, MNG animations, Javascript line graphics, and Unicode text rendering.

WebC can use its own internal EMGL graphics engine for displaying text and images or it can be ported to any graphics display device that provides basic text rendering, bitmap and rectangle drawing, and clipping to rectangular regions. WebC can also optionally support image scaling.

WebC recognizes many character encoding standards, including UTF-8, Unicode 16 bit, Shift-JIS, Chinese BIG5, GB2312, EUCKR Korean, and EUCJP Japanese.

WebC has an offline browsing mode that allows the user to save web content to a memory or disk-based cache for later viewing when the embedded device is not connected to a network. WebC also offers two dimensional tab navigation for TV remote-control style input devices.

Using the WebC API, an application developer can write native function event handlers for HTML or browser-level events, dynamically extend the JavaScript DOM, manipulate HTML documents and HTTP cookies, control screen updates and load queue processing, embed dynamically expanded macros into URLs, recognize custom HTML tag attributes, modify style sheets, and provide custom META tag handling.

Documentation Roadmap

Below is a list of additional WebC documentation available from EBS:

“WebC Porting Guide” – Explains how to port WebC: sockets, SSL, file system, timing, compiler requirements and issues etc. Includes all porting information except that which pertains to the graphics abstraction layer.

“WebC Graphics Abstraction Implementation Guide” – Explains how to implement the Graphics Abstraction Layer.

“WebC C API Guide and Reference” – Details the C Language API for creating WebC applications.

“WebC version x.x Release Notes” – Provides specific information pertaining to each release of WebC.

“RTPlatform Manual” – Complete documentation for the RTPlatform.

WebC Architecture

Figure 1 shows the general structure of a WebC implementation. User applications are written in C or created with HTML and JavaScript. WebC is isolated from platform specifics through a porting layer. This porting layer must be implemented by the implementer or use a porting layer provided by EBS. The porting layer encompasses the following functionality: sockets, SSL, file system, and system timing. The Graphics Layer isolates WebC from the specifics of the graphics subsystem.

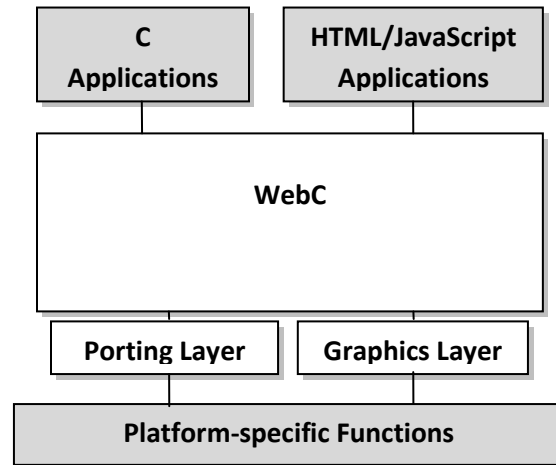


Figure 1

Figure 2 shows a more detailed schematic of WebC. In addition to the porting layer and the graphics abstraction layer. There are a few other significant functional blocks. The RTPlatform contains platform-neutral utility and porting functions. WebC uses the Mozilla SpiderMonkey JavaScript Engine. This subsystem can optionally be omitted to save footprint and memory consumption.

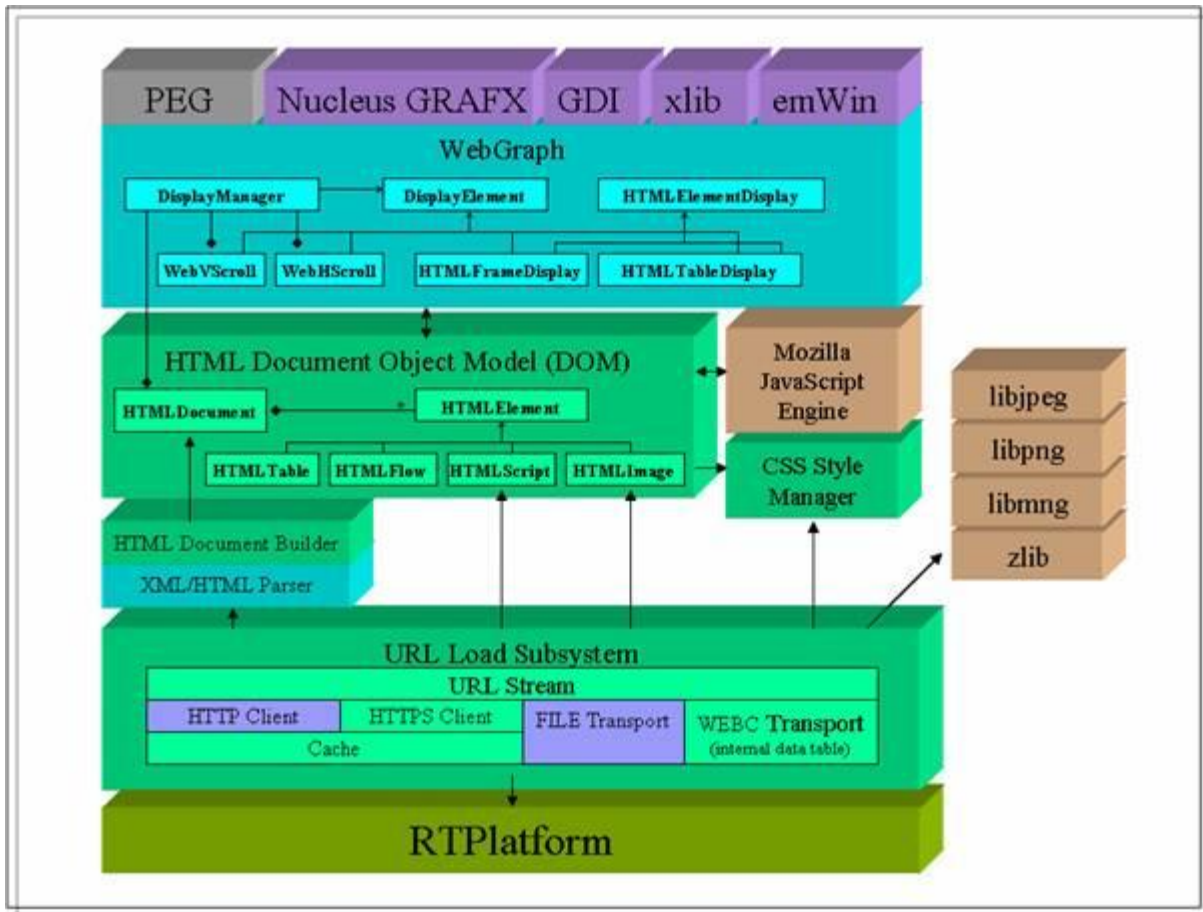


Figure 2

Porting WebC

In general, a complete WebC integration consists of three pieces:

- A port of the relevant RTPlatform modules to the target OS/hardware. RTPlatform is EBS's runtime platform abstraction library. Parts of RTPlatform are portable, or generic, and just implement shared functions or build on other modules. These modules do not need to be ported. Other parts of RTPlatform implement functionality which is specific to the underlying system. These modules must be ported. A specific version of RTPlatform must be linked to the WebC library at build time.
- An implementation of the Graphics Abstraction Layer that acts as the interface between WebC and the low-level graphics library. EBS provides several implementations of this layer or this can be implemented for a specific graphics environment. EBS also provides a generic frame buffer implementation of this abstraction layer.
- Application development using the WebC API and/or HTML and JavaScript.

For complete information on porting WebC, refer to the EBS document titled "WebC Porting Guide".

Simple Example Application

Source Listing 1 shows a simple application that will open a web page. The application creates a new instance of a WebC browser and then opens a specified URL. This example is intended to show how to get a simple application up-and-running with WebC. For a complete understanding of all of the WebC capabilities, see the EBS document titled “WebC C API Guide and Reference”.

```
#include "htmlsdk.h"
#include "webc.h"

const char g_URL[] = "http://www.yahoo.com";

int main()
{
    HTMLBrowserConfig config;
    HBROWSER_HANDLE browser;
    HDOC_HANDLE document;

    // Initialize WebC
    webc_Init();

    // fill the configuration structure by calling the
    //platform-specific graphics layer
    webc_InitPlatformGALBrowserConfig (&config);

    // Now create a browser instance
    config->browserEventHandler = 0;
    config->privateData = 0;
    browser = webc_CreateBrowser(config, WEBC_TRUE, "Explore" , "");
    if(browser)
    {
        // get the document so that we can set the URL
        document = webc_BrowserGetDocument(browser);
        if (document)
        {
            webc_BrowserLoadUrl(browser, g_URL);
        }
    }
    return 0;
}
```

Source Listing 1

Building WebC

The steps to build WebC are, generally, speaking, specific to each platform. Where possible, WebC is designed to be built using command line tools and compilers. This makes it relatively easy to port to other architectures and platforms, as the same makefile can be used with multiple compilers with little modification.

Appendix A provides a detailed overview of the WebC source code directory structure. For each target implementation that EBS provides, there is a in the `.\webc\project` directory. The top-level make files can be found in this directory. For example, the Windows demo build is in

.\\webc\\project\\msvc.net\\webcdemo_gdip and the makefile for the Linux demo is in .\\webc\\project\\linuxgeneric. To build these configurations or to model your implementation after one of these, start with these top-level build directories.

The graphics library implementations that EBS provides are contained in the .\\webc\\gui directory. To use one of these implementations or to construct your own graphics implementation, start with the code in these directories.

Configuring WebC

Compile-time Configuration

The file webcfg.h contains compile time constants to configure web appearance and capabilities. Appendix B shows the various precompiler switches and their meaning.

Run-time Configuration

The file webcdefault.cpp provides initial values of constants that affect webC's appearance. webcdefault.cpp maintains an indexed table of default display values. WebC retrieves a value by calling webc_GetDefault() for that specific value. The default values are initially set by webc_RestoreDefaults() which is called by webc_Init(). The default values may be changed by calling webc_SetDefault() to change individual values.

Appendix A - WebC Source Code Directory Structure

Directory name	Description
manuals	Contains the current WebC manuals
rtplatform\\v.X for Example rtplatform\\v0.4	Container subdirectory for rtplatform directories.
Rtplatform subdirectory contents	
doc	html based documentation
include	Rtplatform include files, this directory must be in your include path.
\\include\\targetname	Target specific include files should reside in subdirectories of include with the same name as

For example: include\vxworks55 include\linux	target.
source	Container subdirectory for source code.
source\generic	Contains Rtpplatform files that can be used on all targets.
source\targetname For example: source\vxworks55 source\linux	Target specific source files reside this directory. Files in this directory provide target specific services such as networking, file IO, and signaling.
webC subdirectory contents	
bin	Tools for compiling HTML content to include into the webC application.
html	Contains "compiled in" HTML, JSCRIPT and image data. These are 'C' files built from HTML and graphics files with an included utility named bintoc.exe. In the Visual C 7.0 project file these C files are built by the project. In other environments these files must be build from the command line. To manually build these files execute the script named html\go.bat from the html subdirectory.
gui	Container subdirectory for gui abstraction code for supported GUI platforms. Not all GUI platforms listed here are supported out of the box, some may require updating before they will run with the latest release of webC.
lib	Container subdirectory containing source code of

	libraries optionally used by webC.
include	All WebC header files referenced from the source are stored here.
project	Each target architecture and compiler has its own project directory.
source	Contains the C and C++ source code modules for WebC
gui subdirectory contents	
Note: Graphics porting sub directories are not all kept up to date. Some are provided as a reference that must be refined. Contact support for more information.	
windows	Windows GDI interface, windows XP
wndml	Wind River windml, (ugl)
xlib	linux/Xwindows
peg	Swell Software's PEG.
pocket_pc	Windows CE on a pocket PC.
Nucleus	Nucleus Graphix
cpeg	Swell Software's CPEG
EMGL	EBS EMGL frame buffer driver
lib subdirectory contents	
lib\libjpeg	Jpeg image decoder library. See: source\imgdecode\webjpeg.cpp
lib\libmng	Mng image decoder library. See: source\imgdecode\webmng.cpp

lib\libpng	Jpeg image decoder library. See: source\imgdecode\webjpeg.cpp
lib\mozilla	"Seamoney" Javascript from Mozilla. Note: Version webC 2.64 and later use version 1.6, earlier versions use version 1.5. Note: To include javascript you must define WEBC_SUPPORT_JSCRIPT from the compiler command line.
lib\openssl	Contains header files and a compiled library derived from SSL, for WindowsXP projects. SSL is no longer directly supported. The best solution is to license 3 rd party SSL solutions.
lib\fdlibm	Math library used by Javascript
lib\zlib	'zlib' general purpose compression library. Used by mnglib
source and include subdirectory contents	
source\browser source\htmldisplay source\htmldom source\imgconvert source\imgdecode source\jscrip source\minigui source\protocol source\util source\webc_example_controls	In version 2.65 and later source code resides in subdirectories of source. In earlier versions all source code resides in the source directory itself.

include\browser	Container subdirectory containing webC core include files. Note: In version 2.65 and later include files reside in subdirectories of include. In earlier versions all include files are in include.
include\htmldisplay	
include\htmlDOM	
include\imgconvert	
include\imgdecode	
include\jscript	
include\minigui	
include\protocol	
include\util	

Appendix B - Precompiler Switches

WEBC_VERSION	Not used currently
WEBC_VERSION_STR	Not used currently
WEBC_OPERATING_SYSTEM	Not used currently
WEBC_USER_AGENT_ID	Used for http
WEBC_SUPPORT_SMALL_SCREEN	Set to 1 to build for a reduced screen device. Default to smaller fonts. Use PDA chrome browser by default. Clip certain display elements instead of displaying horizontal scrollbar.
WEBC_SCREEN_WIDTH	Fixed screen width if WEBC_SUPPORT_SMALL_SCREEN is enabled
WEBC_CFG_STR_FOCUS_SELECT	Set to 1 to select all when a text edit gets the focus and to clear select when it loses focus (traditional). Set to 0 to retain selection if focus changes (useful for soft keypads)
WEBC_CFG_ENABLE_CHROME	Set to 1 to enable HTML-based browser interface

	(requires javascript)
WEBC_SUPPORT_HTTP	Set to 1 to enable support for HTTP: protocol (requires sockets)
WEBC_SUPPORT_FILE	Set to 1 to enable support for FILE: protocol (requires file system)
WEBC_SUPPORT_INTERNAL	Set to 1 to enable the WEBC:// precompiled content with "file names"
WEBC_SUPPORT_UNICODE	<p>This must be defined to support Unicode characters.</p> <p><i>Note: WEBC_SUPPORT_UNICODE is not defined in webcfg.h, it must be defined as a preprocessor variable on the compiler command line.</i></p>
WEBC_SUPPORT_BIG5	Chinese Traditional encoding (requires unicode)
WEBC_SUPPORT_SHIFTJIS	Japanese encoding (requires unicode)
WEBC_SUPPORT_UTF8	Unicode 8-bit transitional format (requires unicode)
WEBC_SUPPORT_EUC_KR	Korean encoding (requires unicode)
WEBC_SUPPORT_EUC_JP	Japanese encoding (requires unicode)
WEBC_CFG_MAX_FONT_FAMILIES	7
WEBC_CFG_HTTP_BUFFER_SIZE	1024
WEBC_CFG_BITMAP_BUFFER_SIZE	1024
WEBC_CFG_MAX_STREAMS	4
WEBC_CFG_META_TABLE_SIZE	1
WEBC_CFG_MAX_ELEMENT_DEPTH	30 // How deep you allow HTML tags to be nested
WEBC_CFG_MAX_FRAME_DEPTH	10 // How deep frames are allowed to be nested
WEBC_DBLCLICK_TIME	300// Max msec time to recognize as a double click event
WEBC_DRAW_IMAGE_BOXES	1

WEBC_DRAW_ALT_TEXT	1
WEBC_MINIMIZE_SCREEN_REFRESH	0
WEBC_USE_3D_FRAMES	1
WEBC_ANIMATE_IMAGES	Set to 1 to enable support for animated images, 0 otherwise.
WEBC_SUPPORT_USER_EVENTS	Set to 1 to enable custom event handlers, 0 otherwise.
WEBC_CFG_MAX_EVENT_HANDLERS	The maximum number of custom user C++ event handlers
WEBC_SUPPORT_STYLE_SHEETS	Set to 1 to enable CSS, 0 otherwise.
WEBC_SUPPORT_TABLES	Set to 1 to enable support for the <TABLE> tag, 0 otherwise
WEBC_SUPPORT_FRAMES	Set to 1 to enable support for HTML frames, 0 otherwise.
WEBC_SUPPORT_IFRAMES	Set to 1 to enable support for inline frames (<IFRAME>), 0 otherwise.
WEBC_SUPPORT_IMAGES	Set to 1 to enable support for the tag, 0 otherwise.
WEBC_SUPPORT_DITHERING	Set to 1 to dither incoming bitmaps
WEBC_SUPPORT_GIFS	Set to 1 to enable support for .GIF image conversion, 0 otherwise. Note: WEBC_SUPPORT_IMAGES must be set to 1 for this option to have any effect.
WEBC_SUPPORT_JPGS	Set to 1 to enable support for .JPG image conversion, 0 otherwise. Note: WEBC_SUPPORT_IMAGES must be set to 1 for this option to have any effect.
WEBC_SUPPORT_PNG	Set to 1 to enable support for .PNG image conversion, 0 otherwise. Note: WEBC_SUPPORT_IMAGES must be set to 1 for this option to have any effect.
WEBC_SUPPORT_MNG	Set to 1 to enable support for .MNG image conversion, 0 otherwise. Note:

	WEBC_SUPPORT_IMAGES must be set to 1 for this option to have any effect.
WEBC_SUPPORT_BMP	Set to 1 to enable support for .BMP image conversion, 0 otherwise. Note: WEBC_SUPPORT_IMAGES must be set to 1 for this option to have any effect.
WEBC_SUPPORT_BACKGROUND_IMAGES	Set to 1 to enable support for background images, 0 otherwise. Note: Only has effect if WEBC_SUPPORT_IMAGES=1.
WEBC_SUPPORT_ZINDEX	Set to 1 to enable handling for z-positioned elements, 0 to save a little space
WEBC_SUPPORT_RELATIVE_POS	Set to 1 to enable support for relative-positioned elements
WEBC_SUPPORT_SPATIAL_TAB_NAVIGATION	Set to 1 to enable support for spatial navigation in forms
WEBC_SUPPORT_INNER_HTML	Set to 1 to enable support for innerhtml and outerhtml properties. Inner html support requires additional memory for each element. Embedded applications that do not need these features can reduce memory consumption by disabling.
WEBC_SUPPORT_CACHE	Set to 1 to support caching, 0 otherwise. Uses a file based caching mechanism for text and images. File system calls are abstracted through Rtplatform. See webcfile.c See also : WEBC_CACHE_PAGE_SOURCE and WEBC_CACHE_IMAGES
WEBC_SUPPORT_COOKIES	Set to 1 to enable cookies, 0 otherwise.
WEBC_SUPPORT_OFFLINE_BROWSING	Set to 1 to enable the offline content cache
WEBC_SUPPORT_URL_MACROS	Set to 1 to enable URL Macros
WEBC_SUPPORT_STORE_BITMAP	Set to 1 to enable code for store and retrieving images (required for image caching)
WEBC_SUPPORT_URL_LOG	Set to one to write all urls visited to a log file
WEBC_CFG_URL_LOG_FILE	The log file name used when WEBC_SUPPORT_URL_LOG is enabled

WEBC_CFG_HISTORY_SIZE	Number of URLs stored in the history buffer
WEBC_CFG_EVENT_QUEUE_SIZE	Size of HTMLEvent queue
WEBC_CFG_EVENT_MAX_RECURSE_DEPTH	Max recursion depth for HTML::Event
WEBC_CFG_MAX_COOKIE_BYTES	Maximum cookie size, not used if WEBC_SUPPORT_COOKIES is 0
WEBC_CONVERT_FW_ASCII_TO_ASCII	Automatically convert Unicode fullwidth ascii characters to normal ascii (only relevant when WEBC_SUPPORT_UNICODE is enabled)
WEBC_SUPPORT_HTTPS	Enable and configure HTTPS support
WEBC_SUPPORT_VERIFY	Note: HTTPS source code is included but technical support for HTTPS and source code and
HTTPS_CERT_FILE	technical support for SSL is no longer provided
WEBC_CACHE_MAX_PATH_LEN	128
WEBC_CFG_CACHE_PREFIX	"cache\\"
WEBC_CFG_OFFLINE_CACHE_PREFIX	"archive\\"
WEBC_CFG_BMP_CACHE_PREFIX	WEBC_CFG_CACHE_PREFIX "bmp\\"
WEBC_CFG_ONLINE_CACHE_SIZE	(1024L*1024L*100L)
WEBC_CFG_ONLINE_CACHE_MAX_FILES	1000
WEBC_CFG_OFFLINE_CACHE_SIZE	(1024L*1024L*100L)
WEBC_CFG_OFFLINE_CACHE_MAX_FILES	10000
WEBC_CACHE_IMAGES	Set to 1 to cache binary image data for images, 0 otherwise. The caching is done after the image has been converted, for best performance. This is meaningful only when WEBC_SUPPORT_CACHE=1.
WEBC_CFG_HTTP_CACHE_LOCATION	where to store cached files
WEBC_CFG_HTTP_CACHE_SIZE	The maximum bytes to use for caching HTTP data

WEBC_CFG_HTTP_CACHE_MAX_FILES	The maximum number of files in the HTTP cache
WEBC_HTTP_USE_RAM_CACHE	define this symbol to use a RAM-based cache for HTTP
WEBC_CFG_CACHE_IMG_EXPIRE_TIME	The default image expire time = 3 min
WEBC_NO_TABLE_AUTO_CLEAR	Set to 1 to be compatible with Mozilla & IE; set to 0 for better formatting on narrow displays
WEBC_SUPPORT_JSCRIPT	<p>This must be defined to support Javascript.</p> <p><i>Note: WEBC_SUPPORT_JSCRIPT is not defined in webcfg.h, it must be defined as a preprocessor variable on the compiler command line.</i></p>
WEBC_SUPPORT_BODY_EVENTS	Turn this on if you want to body to receive onClick, onMouseover etc... events NOTE: This only works if JScript is enabled and it will hurt performance.
WEBC_CFG_JS_CX_POOL_SIZE	lets be generous
WEBC_APP_CODE_NAME	"Mozilla"//use these defines to be recognized as MSIE 6.0
WEBC_APP_NAME	"Microsoft Internet Explorer"
WEBC_APP_MINOR_VERSION	"0"
WEBC_APP_VERSION	"4.0 (compatible; MSIE 6.0; Windows 98; COM+ 1.0.2204)"
WEBC_APP_CODE_NAME	"WebC" // Use these defines to be recognized as WebC
WEBC_APP_NAME	"EBS WebC"
WEBC_APP_MINOR_VERSION	"0"
WEBC_APP_VERSION	WEBC_VERSION_STR " (compatible; WebC " WEBC_VERSION_STR "; Windows 98;)"
WEBC_CPU_CLASS	"x86" //these defines should be set to reflect the platform and environment that WebC is running on

WEBC_PLATFORM	"Win32"
WEBC_SYSTEM_LANGUAGE	"en-us"
WEBC_USER_LANGUAGE	"en-us"
WEBC_MESSAGEFLOW_DEBUG	Set to 1 to enable support for viewing internal message flow. If enabled, htmldiag.cpp contains code to print the message flow to HTML elements, and underlying HTML Display Elements Tables contained in the file allow you to select which messages to which elements should be displayed.
WEBC_MEMORY_DEBUG	If set to 1, internal counters will be maintained for each call to malloc and free. Tools are available to find memory leaks.
WEBC_LOAD_BUFFER_SIZE	8192
WEBC_TIMEOUT_SEC	Number of seconds to wait for response from server either when connecting or downloading data.
WEBC_DEFAULT_INPUT_SIZE	The default <SIZE> value for <INPUT> tags, i.e. the maximum number of visible characters.
WEBC_DEFAULT_INPUT_MAX	The default <MaxLength> value for <INPUT> tags, i.e. the maximum number of characters that can be entered into the text box.
WEBC_TAB_SIZE	This determines the tab size, in spaces.
WEBC_HR_HEIGHT	The height, in pixels, of a horizontal rule if not specified as a parameter in the HTML page (<HR> tag).
WEBC_HR_SPACING	The space, in pixels, to put above and below a horizontal rule.
WEBC_LINE_SPACING	This controls the number of blank pixels in between lines.
WEBC_BROKEN_IMG_LINK_WIDTH	control the width and height, respectively, for tags with no WIDTH or HEIGHT attributes whose SRC files cannot be
WEBC_BROKEN_IMG_LINK_HEIGHT	opened, i.e. "broken links".

WEBC_DEFAULT_TEXTAREA_COLS	Determine the default dimensions of a <TEXTAREA> element.
WEBC_DEFAULT_TEXTAREA_ROWS	2
WEBC_CFG_DEFAULT_TEXT_HEIGHT	Default values for input style ??
WEBC_CFG_DEFAULT_VSCROLL_STEP	32
WEBC_CFG_DEFAULT_HSCROLL_STEP	32
WEBC_CFG_BULLET_WIDTH	12
WEBC_CFG_BULLET_HEIGHT	16
WEBC_CFG_DEFAULT_LANGUAGE	"en-us,en;q=0.5"
WEBC_SUPPORT_HTTP_PROXY	These constants control the HTTP proxy features of webc
WEBC_HTTP_PROXY_ENABLE	not supported
WEBC_HTTP_PROXY_PORT	6588
WEBC_CFG_MAX_LOAD_NESTING	how deep HTMLBrowser loads can be nested

Appendix C - Run-time Configuration

Values managed by WEBCDEFAULT.CPP

WEBC_DEFAULT_ACTIVE_BACKGROUND_COLOR
WEBC_DEFAULT_ACTIVE_COLOR
WEBC_DEFAULT_ACTIVE_LINK_COLOR
WEBC_DEFAULT_BACKGROUND_COLOR
WEBC_DEFAULT_BODY_FONT_FAMILY
WEBC_DEFAULT_BODY_FONT_SIZE
WEBC_DEFAULT_BODY_FONT_STYLE
WEBC_DEFAULT_BODY_FONT_WEIGHT

WEBC_DEFAULT_BODY_PADDING
WEBC_DEFAULT_BUTTON_BACKGROUND_COLOR
WEBC_DEFAULT_BUTTON_FONT_FAMILY
WEBC_DEFAULT_BUTTON_FONT_SIZE
WEBC_DEFAULT_BUTTON_FONT_STYLE
WEBC_DEFAULT_BUTTON_FONT_WEIGHT
WEBC_DEFAULT_BUTTON_FRAME_WIDTH
WEBC_DEFAULT_BUTTON_HILITE_COLOR
WEBC_DEFAULT_BUTTON_SHADOW_COLOR
WEBC_DEFAULT_BUTTON_PADDING
WEBC_DEFAULT_HILITE_COLOR
WEBC_DEFAULT_LINK_COLOR
WEBC_DEFAULT_LIST_INDENT
WEBC_DEFAULT_SCROLL_BAR_BUTTON_LENGTH
WEBC_DEFAULT_SELECT_FONT_FAMILY
WEBC_DEFAULT_SELECT_FONT_SIZE
WEBC_DEFAULT_SELECT_FONT_STYLE
WEBC_DEFAULT_SELECT_FONT_WEIGHT
WEBC_DEFAULT_SHADOW_COLOR
WEBC_DEFAULT_SLIDER_BACKGROUND_COLOR
WEBC_DEFAULT_SLIDER_COLOR
WEBC_DEFAULT_SLIDER_FRAME_WIDTH
WEBC_DEFAULT_SLIDER_BACKGROUND_FRAME_WIDTH
WEBC_DEFAULT_SLIDER_HILITE_COLOR

WEBC_DEFAULT_SLIDER_PADDING
WEBC_DEFAULT_SLIDER_SHADOW_COLOR
WEBC_DEFAULT_SLIDER_WIDTH
WEBC_DEFAULT_SLIDER_FRAME_SHADOW_COLOR
WEBC_DEFAULT_SLIDER_FRAME_HILITE_COLOR
WEBC_DEFAULT_TEXT_AREA_FONT_FAMILY
WEBC_DEFAULT_TEXT_AREA_FONT_SIZE
WEBC_DEFAULT_TEXT_AREA_FONT_STYLE
WEBC_DEFAULT_TEXT_AREA_FONT_WEIGHT
WEBC_DEFAULT_TEXT_COLOR
WEBC_DEFAULT_TEXT_INPUT_FONT_FAMILY
WEBC_DEFAULT_TEXT_INPUT_FONT_SIZE
WEBC_DEFAULT_TEXT_INPUT_FONT_STYLE
WEBC_DEFAULT_TEXT_INPUT_FONT_WEIGHT
WEBC_DEFAULT_VISITED_LINK_COLOR
WEBC_DEFAULT_SCROLL_CORNER_COLOR