Integrating Palm OS® Applications with Web Browser

Palm OS® 5 SDK Web Browser Update

CONTRIBUTORS

Written by Susan Salituro
Engineering contributions by Mohomed Najeeb Abdulrahiman, Brian Chin, Kathy Ly, Bob Whiteman.

PalmSource, Inc. reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of PalmSource, Inc. to provide notification of such revision or changes.

PALMSOURCE, INC. AND ITS SUPPLIERS MAKE NO REPRESENTATIONS OR WARRANTIES THAT THE DOCUMENTATION IS FREE OF ERRORS OR THAT THE DOCUMENTATION IS SUITABLE FOR YOUR USE. THE DOCUMENTATION IS PROVIDED ON AN "AS IS" BASIS. PALMSOURCE, INC. AND ITS SUPPLIERS MAKE NO WARRANTIES, TERMS OR CONDITIONS, EXPRESS OR IMPLIED, EITHER IN FACT OR BY OPERATION OF LAW, STATUTORY OR OTHERWISE, INCLUDING WARRANTIES, TERMS, OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND SATISFACTORY QUALITY. TO THE FULL EXTENT ALLOWED BY LAW, PALMSOURCE, INC. ALSO EXCLUDES FOR ITSELF AND ITS SUPPLIERS ANY LIABILITY, WHETHER BASED IN CONTRACT OR TORT (INCLUDING NEGLIGENCE), FOR DIRECT, INCIDENTAL, CONSEQUENTIAL, INDIRECT, SPECIAL, OR PUNITIVE DAMAGES OF ANY KIND, OR FOR LOSS OF REVENUE OR PROFITS, LOSS OF BUSINESS, LOSS OF INFORMATION OR DATA, OR OTHER FINANCIAL LOSS ARISING OUT OF OR IN CONNECTION WITH THIS DOCUMENTATION, EVEN IF PALMSOURCE, INC. OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Integrating Palm OS® Applications with Web Browser
Document Number 9002-001
August 30, 2002
For the latest version of this document, visit
http://www.palmos.com/dev/support/docs/.

PalmSource, Inc.
1240 Crossman Ave.
Sunnyvale, CA 94089
USA
www.palmsource.com

1

Integrating Palm OS® Applications with Web Browser

The following sections describe a few ways in which your applications can take advantage of certain features of PalmSource® Web Browser version 2.0:

- "Opening a Web Page from an Application" on page 1
- "Opening the Bookmark List from an Application" on page 4
- "Changing the Home Page Setting from an Application" on page 4
- "Registering Applications with Exchange Manager" on page 7
- "Launching an Application from a Web Page" on page 8

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.

IMPORTANT: Web Browser version 2.0 only runs on handhelds that use Palm OS 5 or later. See http://www.palmsource.com for the latest Palm OS SDK.

Opening a Web Page from an Application

You can either launch or sublaunch Web Browser from a Palm OS application by using the `sysAppLaunchCmdGoToURL` launch code.

NOTE: Older versions of Web Browser (known as the Web Clipping Application Viewer or Clipper) also used the `sysAppLaunchCmdGoToURL` launch code for launching Web Browser. However, Clipper did not support sublaunching web pages from an application.

When you launch Web Browser from another application, consider these two cases:

- Web Browser is not the active application. In this case, Palm OS will launch Web Browser using the `SysUIAppSwitch` function. Web Browser will process the request once it is loaded.
- Web Browser is the active application. (This case occurs when Web Browser sublaunched another application, and that application made the request.) In this case, Web Browser will memorize the sublaunch request without executing it and then return to the calling application. The calling application must return to Web Browser in order for Web Browser to resume operation. Once control is returned to Web Browser, it will execute the request. This behavior lets external applications control browser navigation without giving up Web Browser features such as the Back button and the History list.

The example in Listing 1.1 shows how to write a function that will launch or sublaunch Web Browser to its Web Content dialog with the appropriate arguments.

Listing 1.1 WebBrowserCommand example

```
#include <PalmOS.h>
#include <WebBrowser.h>

Err  WebBrowserCommand(Boolean subLaunch, 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, WebBrowserCreator, true, &cardNo, &dbID);

  if (error == errNone)
  {
    if (subLaunch)
    {
      SysAppLaunch(cardNo, dbID, launchFlags, command, parameterP, &result);
      if (resultP)  *resultP = result;
    }

    else
    {
      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 Web Browser from your application, call this function using a command similar to the following example:

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

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

To open a specific web page by sublaunching Web Browser, use the same command, but change the value of the `subLaunch` parameter to `true`:

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

See the AppLaunchWeb sample application in the Palm OS 5 version 5.1 SDK Web Browser Update for an example that uses this launch code.

Opening the Bookmark List from an Application

If your application includes a function like the one in Listing 1.1, use a command like the following example to launch Web Browser to its Bookmark list dialog:

```
WebBrowserCommand(false, 0, sysAppLaunchWebBrowserLaunchBookmarkDialog, NULL,
NULL);
```

If the user does not select any bookmark and taps **Done** in the Bookmark list dialog, the main Web Browser window will appear.

NOTE: Sublaunching Web Browser to the Bookmark list dialog is not supported.

See the AppLaunchWeb sample application in the Palm OS SDK Web Browser Update for an example that uses this launch code.

Changing the Home Page Setting from an Application

A Palm OS application may sublaunch Web Browser to get or modify its home page setting. On its next launch, Web Browser will detect the changed setting and launch the new home page.

Two launch codes, `sysAppLaunchWebBrowserGetHomePage` and `sysAppLaunchWebBrowserSetHomePage`, let you get and set the home page from an application. You can only use these launch codes when your application sublaunches Web Browser (using SysAppLaunch).

The example in Listing 1.2 illustrates how you might use these codes to modify Web Browser's home page setting.

Listing 1.2 Example of changing the home page setting

```
#include <PalmOS.h>
#include <WebBrowser.h>
{
  DmSearchStateType          searchState;
  UInt16                     cardNo;
  LocalID                    dbID;
  Err                        error;
  WebBrowserHomePageAPIType  homePage;
  UInt32                     wbRetValue;

  error = DmGetNextDatabaseByTypeCreator( true, &searchState,
  sysFileTApplication, WebBrowserCreator, true, &cardNo, &dbID);

  if (error == errNone)
  {
    MemSet( &homePage, sizeof(homePage), 0 );
    error = SysAppLaunch( cardNo, dbID,
              0,                  // launchFlags == 0
              sysAppLaunchWebBrowserGetHomePage,
              &homePage,
              &wbRetValue);       // Value returned by Web Browser

    if ((error == 0) && (homePage.homePageURLP != NULL))
    {  // We got the Home Page URL
      MemPtrFree(homePage.homePageURLP);
      // Free memory allocated by Web Browser
if (homePage.modifiedByUser == 0) // Was never changed by Web Browser
        {
          #define newHomePage "file:MyWCHomePage.prc"
          homePage.homePageURLP = MemPtrNew( sizeof(newHomePage) );
          // Includes zero term.

          if (homePage.homePageURLP)
          {
            StrCopy( homePage.homePageURLP, newHomePage );
            homePage.modifiedByUser = 2;// was changed for this particular app
            error = SysAppLaunch( cardNo, dbID,
                      0,                // launchFlags == 0
                      sysAppLaunchWebBrowserSetHomePage,
                      &homePage,
                      &wbRetValue);     // Value returned by Web Browser

            if (error == 0)
            {
               FrmAlert( NewHomePageSuccessAlert );
            }
            MemPtrFree( homePage.homePageURLP );
          }
        }
     }
   }
}
```

See the AppLaunchWeb sample application in the Palm OS SDK Web Browser Update for an example that uses these launch codes.

TIP: By default, Web Browser assumes the calling application is running in the PACE emulator (68K mode). If the calling application is ARM-native (not running in PACE), you need to set the `flags` field of the `WebBrowserHomePageAPIType` structure to the value `1`. See the header file `WebBrowser.h` in the Palm OS SDK Web Browser Update for details about this structure's other fields.

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.

Handling Web Content

Web Browser itself is registered to handle the following content (MIME) types: `text/html`, `image/gif`, and `image/jpeg`. It is registered with Exchange Manager for handling these types, but not as the "default" application, so it can be overridden by another application.

Handling Other Content Types

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, Listing 1.3 shows how to register a mail application to handle the `mailto:` scheme.

Listing 1.3 Registering a Mail Application

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

When you register a new email-handling application, 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 Web Browser encounters the `mailto:` scheme.

For another example that uses Exchange Manager to send content to another application for display, see the WebExg example in the Palm OS SDK Web Browser Update.

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 Web Browser.

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=PalmSource, Inc.&city=Sunnyvale">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.