



Creating Multi-lingual Web Pages Using the Advanced Web Server

Introduction

The Advanced Web Server is a component of NetSilicon's NET+OS embedded operating system. The Advanced Web Server allows you to change the language in which pages are served through the use of phrase dictionaries. This document explains the use of phrase dictionaries and why you should use this feature for all pages that will or might be translated into multiple languages.

The Advanced Web Server and its pbuilder utility are premier products from NetSilicon that allow you to create and deliver highly sophisticated WebWeb pages to your customers. Through comment tags, you can include variable data in Web pages along with static page data. NetSilicon recommends that you use phrase dictionaries in combination with your HTML code. With this method, you use the Advanced Web Server and its pbuilder utility as they were designed to be used.

A phrase dictionary is a file (RpUsrDct.txt) that contains phrases that represent static HTML data. The phrases are then encoded into two-byte codes (tokens). The pbuilder utility converts your HTML files into C source code and finds phrases in your HTML file that match the phrases in RpUsrDct. The pbuilder utility, as it writes out your C source code equivalent files, replaces phrases in the HTML file with two-byte codes in the resultant C source code file. When your pages are then served to a browser, instead of directly using phrases, the two-byte code is used to index into an array of phrases, in whatever language you selected.

To use these methods, perform the steps listed here:

1. Create your Web pages, including comment tags, as you normally do.
Add all static text included in your Web pages to the RpUsrDct.txt file. In addition, ensure that the phrases added to the RpUsrDct.txt file are the same as those included in your Web pages, as pbuilder uses the phrases from RpUsrDct.txt as hashes while searching your Web pages.
2. Run the pbuilder utility against your Web pages.
The pbuilder utility creates two additional files, besides the .c and the _v.c files that define your Web pages: RpUsrDct.c and RpUsrDct.h.
3. Include RpUsrDct.h in your .c file or files that the pbuilder utility created.
4. For additional language support, have file RpUsrDct.c (generated by the pbuilder utility) translated into the languages that your application supports. You can rename the additional RpUsrDct.c files according to the language they support; for example, RpUsrDctGer.c (German), RpUsrDctFre.c (French), and so on.
5. Include the additional RpUsrDct*.c files in your Green Hills MULTI build or GNU make files as you would any additional .c files.

6. When your application is executing, use this call to change the language in which your pages will be displayed:

```
RpSetUserPhraseDictionary(ServerData, Dictionary,0)
```

where

- ServerData is a pointer to void and points to the engine internal data structure. In some of your code, a pointer to this structure may already be available to you. You also can access it by calling RpHsGetServerData.
- Dictionary is the array of the phrases contained in the RpUsrDct*.c file that you had translated.
- 0 is a zero that represents no decompression of the contents of the dictionary.

The call to RpSetUserPhraseDictionary might be called in a stub function as the result of a user selecting a language from a Web page that you create in some default language.

Additionally, 1024 two byte tokens are available in user phrase dictionaries. The first 64 are reserved for RomPager engine error messages. The remainder is available for user phrase dictionary entries.

Creating Web Pages When the Starting Language is not English

Many customers create their application in a language other than English, and then translate the pages into English along with other languages. Follow the instructions listed next (they are similar to the steps listed in the previous section).

1. Create your HTML pages in your native language.
2. Place all phrases to be translated into the file RpUsrDct.txt. Run the HTML 1 pages(s) (and RpUsrDct.txt) through pbuilder.
3. Place the resulting phrases contained in the RpUsrDct.c file translated into the additional languages as required.
4. Create multiple RpUsrDct.c files whose names reflect the languages their phrases are written in.
5. Include these .c files in the .bld or make files as required.

It is crucial that you call RpSetUserPhraseDictionary just after you start up the Web server. Even if the language you want displayed is the default language in which you wrote the application, you must call RpSetUserPhraseDictionary to establish the phrase dictionary in use for the application.

In conclusion, NetSilicon strongly recommends using phrase dictionaries as the method for including phrases in Web pages and for making provisions for adding multi-lingual Web pages. This is the supported method and leaves provisions for translating phrases to multiple languages.