

Creating External Libraries for Corel's Personal Web Server

Corel's Personal Web Server is a light weight World Wide Web (or HTTP) server. Web servers on Unix and other platforms, and some Windows servers, support a method of allowing the user to create programs or scripts which will be called when the user requests that page. CGI, or Common Gateway Interface, is the name typically given to this.

Personal Web Server solves this problem in a different, and more flexible, manner. In PWS, you can create extension DLLs for the server which will have the opportunity of replacing text in the HTML document as it is delivered to the client.

For example, if you wanted to create a web page that would return employee phone numbers when given employee names, you would use a tag like this:

```
The phone number is <extdll phone>.
```

An external DLL would replace the text within the tag, with the requested phone number. The DLL has access to any form data submitted, and any of the header fields from the user's web browser.

There is no limit to the number of <extdll> tags you can have in a document, nor to the amount of text a DLL can return to substitute for one tag. DLL's also don't incur the penalty of creating a new process every time the web page is accessed.

The Mechanism

When the server is started, it looks in the following registry key:

```
HKEY_CURRENT_USER\Software\Corel\Internet Mania\WebServe\External
```

Where it expects to find a name/value pair for each external library. The name will be compared with the name given in the <extdll> tag and the value is the full path to the DLL which will be used. For example, a name of "phone" and a value of "c:\external\phone.dll" would cause the <extdll phone> tag to call a function in c:\external\phone.dll.

The Implementation

The DLL name must have an entry point named "ProcessRequest", which must accept a C style function call with the following prototype:

```
HGLOBAL ProcessRequest(const char *pszHeader,  
                        const char *pszArgs, SOCKET hSocket);
```

The ProcessRequest() function will be called once per <extdll> tag, and returns a global memory handle containing the data to be sent to replace the tag. A NULL handle will insert no data. The size of the global memory object will be the number of bytes sent.

The pszHeader argument is a pointer to the HTTP header that PWS retrieved from the start of the document. Each line of the header will be separated by a carriage-return/linefeed combination, so the string you get will look like this:

```
GET /headertest.html HTTP/1.0  
Accept: /*/*, image/x-xbitmap, image/jpeg, image/gif  
Accept-Language: en  
User-Agent: Microsoft Internet Explorer/4.40.474beta (Windows 95)  
Connection: Keep-Alive
```

Usually you only need the first line of the request, but there are uses for the rest of the header information so it is provided. If the request is a form submission, then the first line of the header might look like this:

```
GET /headertest.html?name=frank HTTP/1.0
```

Notice the Name field of the form is right on the first line of the header. For more information on the format of the data you'll get here, consult the HTML specification.

The second argument to `ProcessRequest()` is the rest of the `<extdll>` tag. For example, if the tag looked like this: `<extdll yourlib type=3>` then the `pszArgs` field would point to the string "type=3".

The final argument is the actual socket handle that the user's web browser is connected to on the other end. You can write data directly to the socket if you like, but it is recommended that you return the data in the global memory object. If you have a huge file to write, for example, you may choose to send it yourself. You can also use the socket handle to get the IP address of the other end of the connection.

The Example

Using Visual C++, use the AppWizard to create a new MFC DLL project. Add the following code to the C++ code that it generates:

```
extern "C" __declspec(dllexport) HGLOBAL
ProcessRequest(const char *pszHeader, const char *pszArgs, SOCKET hSocket)
{
    // Build a Hello World string in a global memory object
    HGLOBAL hGlob = GlobalAlloc(GHND, 11);
    if (hGlob)
    {
        char *p = (char *)GlobalLock(hGlob);
        if (p)
        {
            memcpy(p, "Hello World", 11);
            GlobalUnlock(hGlob);
        }
    };

    // the server will free the memory once it's been sent
    return hGlob;
}
```

Compile and link this library, add it to the registry, create an HTML document with the proper `<extdll>` tag in it, drop the document into the WebServe window, and double-click on it. You should see "Hello World" where your `<extdll>` directive was.