Libwww for OpenVMS IA64

Brett Cameron (brett.cameron@hp.com) and John Apps (john.apps@hp.com)

18 November 2008

Disclaimer: the information in this document is the sole responsibility of the authors. The information contained herein is offered on a best-effort basis. Hewlett-Packard offers no support or warranty on any of the content in this document or the software described in it.

1. Introduction

Thank your for your interest in this port of W3C `libwww` for OpenVMS Alpha and Integrity (IA64).

`Libwww` (see http://www.w3.org/Library/) is a general-purpose Web API written in C. With a highly extensible and layered API, it can accommodate many different types of applications including clients, robots, and so on. The purpose of `libwww` is to provide a highly optimized HTTP sample implementation as well as other Internet protocols and to serve as a test-bed for protocol experiments. `Libwww` also supports HTTPS through OpenSSL.

The basic design model can be viewed here. An architectural diagram is shown in Figure 1 Libwww Design Model.

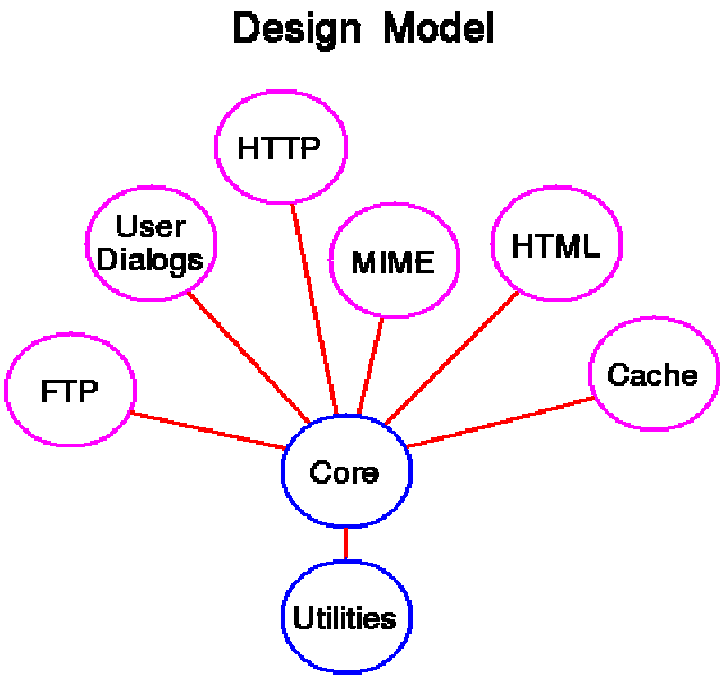

Figure 1 Libwww Design Model

Web applications built using `libwww` included in this release are a robot (`webbot.exe`) and a useful command line tool (`w3c.exe`). The robot can crawl Web sites faster, and with lower load, than many other Web walkers, due to its extensive pipelining and use of HTTP/1.1. The command line tool is very useful for manipulation of Web sites that implement more than just HTTP `GET` (`PUT`, `POST`, and so on).

This port of `libwww` for OpenVMS leverages previous porting efforts for OpenVMS Alpha and is a more complete implementation of the API. That being said, the library may not be 100% complete, and we have not yet performed sufficient testing to guarantee that all features of the API work correctly. Accordingly, if you do notice anything that is missing from the library or find functions that are not working correctly, please let us know and we will endeavour to fix the problem.

This work has been done in the hope that people will find it useful. The authors have utilised this API a number of times over the past 8 or 9 years on various Linux and UNIX-based projects, and have found it to be very robust and effective.

2. Requirements

The kit you are receiving has been compiled and built using the operating system and compiler versions listed below. While it is highly likely that you will have no problems installing and using the kit on systems running higher product versions of the products listed, we cannot say for sure that you will be so lucky if your system is running older versions. If you would like a kit for a different configuration (maybe even for VAX), we will do what we can to oblige. Note that the UnZip utility is required to unpack the ZIP kit.

- OpenVMS 8.3 IA64
- HP TCP/IP Services for OpenVMS Industry Standard 64 Version V5.6
- HP C S7.1-013
- UnZip 5.42 (or similar) for OpenVMS (required to unpack the kit)

In addition to the above requirements, it is assumed that the reader has a good knowledge of OpenVMS and of software development in the OpenVMS environment.

2.1. Optional products

While `libwww` was implemented to facilitate the development of Web applications using C, language integration issues aside, under OpenVMS in particular there is no particular reason why it cannot be used to develop Web applications using practically any OpenVMS 3GL you can think of, such as COBOL, BASIC, Pascal, or FORTRAN. However, it should be noted that regardless of your language preferences, it will invariably be necessary to write at least some C code to take care of language incompatibilities.

3. Recommended reading

Before getting too carried away, be sure to read (or at least browse) the documentation provided on the W3C Web site (http://www.w3.org/Library/User/).

4. Contents of the kit

The following file listing represents the output from the `unzip -l` command run against the ZIP file kit. As you can clearly see, there is not that much to the distribution; however what is there is all that you need to do most of what is described in the W3C Web pages (http://www.w3.org/Library/User/)[1].

```
Archive:  LIBWWW-VMS-I64-01.ZIP;1
  Length      Date    Time    Name
 --------     ----    ----    ----
        0  11-12-08 23:08   libwww/bin/
     1694  11-15-08 20:09   libwww/copyright.txt
        0  11-12-08 23:09   libwww/include/
        0  11-12-08 23:08   libwww/lib/
      261  11-15-08 11:43   libwww/libwww$startup.com
        0  11-12-08 23:09   libwww/samples/
  2299904  11-15-08 11:49   libwww/bin/libwww.exe
    40448  11-18-08 03:34   libwww/bin/w3c.exe
   125952  11-18-08 03:34   libwww/bin/webbot.exe
```

[1] The source code for this build is provided in the ZIP file `LIBWWW-VMS-SRC-01.ZIP`, which may be downloaded from the same location as the runtime kit.

```
   725504  11-15-08 11:49   libwww/include/libwww.tlb
  3198464  11-15-08 11:45   libwww/lib/libwww.olb
     2051  11-15-08 11:41   libwww/samples/chunk.c
     3795  11-15-08 11:41   libwww/samples/chunkbody.c
     4760  11-15-08 11:41   libwww/samples/cookie.c
    27734  11-15-08 11:41   libwww/samples/davsample.c
     4413  11-15-08 11:41   libwww/samples/eventloop.c
     3549  11-15-08 11:41   libwww/samples/getheaders.c
     2801  11-15-08 11:41   libwww/samples/head.c
     2212  11-15-08 11:41   libwww/samples/init.c
     3555  11-15-08 11:41   libwww/samples/isredirected.c
      105  11-15-08 11:41   libwww/samples/libapp_1.c
      856  11-15-08 11:41   libwww/samples/libapp_2.c
      950  11-15-08 11:41   libwww/samples/libapp_3.c
     1496  11-15-08 11:41   libwww/samples/libapp_4.c
     5163  11-15-08 11:41   libwww/samples/listen.c
     3632  11-15-08 11:41   libwww/samples/loadtofile.c
     3310  11-15-08 11:41   libwww/samples/memput.c
     3717  11-15-08 11:41   libwww/samples/mget.c
     3697  11-15-08 11:41   libwww/samples/multichunk.c
    22502  11-15-08 11:41   libwww/samples/myext.c
    29924  11-15-08 11:41   libwww/samples/myext2.c
     3324  11-15-08 11:41   libwww/samples/post.c
     3221  11-15-08 11:41   libwww/samples/postform.c
     4165  11-15-08 11:41   libwww/samples/ptri.c
     2967  11-15-08 11:41   libwww/samples/put.c
     4079  11-15-08 11:41   libwww/samples/range.c
     4209  11-15-08 11:41   libwww/samples/rdf_parse_buffer.c
     1974  11-15-08 11:41   libwww/samples/rdf_parse_file.c
     4656  11-15-08 11:43   libwww/samples/samples.com
     3384  11-15-08 11:41   libwww/samples/showlinks.c
     4186  11-15-08 11:41   libwww/samples/showtags.c
     2933  11-15-08 11:41   libwww/samples/showtext.c
     9091  11-15-08 11:41   libwww/samples/showxml.c
     3142  11-15-08 11:41   libwww/samples/stri.c
    13204  11-15-08 11:41   libwww/samples/tiny.c
     3181  11-15-08 11:41   libwww/samples/trace.c
     1052  11-15-08 11:41   libwww/samples/tzcheck.c
     2951  11-15-08 11:41   libwww/samples/upgrade.c
     4084  11-15-08 11:41   libwww/samples/wwwssl.c
--------                   -------
  6598252                   49 files
```

Note that all of the C header files required to use the API have been placed into a text library (`libwww.tlb`) that needs to be included when you compile any code. Alternatively, you may wish to extract the contents of the text library and specify an appropriate include path for your compile operations.

It should also be noted that HTTPS support (SSL via OpenSSL) is only included in the object library (`libwww.olb`), and is not included in the shareable image (`libwww.exe`). Consequently, if you wish to use HTTPS, it will be necessary to link your code with the object library as opposed to the shareable image.

5. Installation

Unpacking and installing the ZIP file kit is very straightforward. After copying the ZIP file (`LIBWWW-VMS-I64-01.ZIP`) to a suitable location, simply enter the following command to unpack the contents of the ZIP file. After unpacking the kit, you will have a `[.libwww]` directory in your current directory that contains files listed above.

```
$ unzip LIBWWW-VMS-I64-01.ZIP
```

5.1. Post-installation steps

To complete the installation, it is necessary to define the logical name `libwww$root` to point to your top-level `[.libwww]` directory, and to define a logical name for the shareable image (`libwww.exe`). In addition, you may want to define a logical name for the text library containing the C header files (`libwww.tlb`). A sample command procedure called `libwww$startup.com` that illustrates these logical name definitions is provided in the top level `libwww` directory (see listing below). You may wish to modify this file as required and include it in your system start-up procedure.

```
$ set noon
$
$ ! Modify definition for libwww$root as appropriate
$ define/system/trans=concealed libwww$root dorone$dka0:[libwww.]
$
$ define/sys/exec libwww libwww$root:[bin]libwww.exe
$ define/sys/exec libwww$include libwww$root:[include]libwww.tlb
$
$ exit
```

5.2. Privileges and quotas

Generally speaking, there are no special quota or privilege requirements for applications developed using the `libwww` port, although a reasonably high `BYTLM` is recommended. The authors typically operate (on OpenVMS IA64) with quota settings similar to the following, which should be more than adequate for most purposes:

```
Maxjobs:         0  Fillm:      4096  Bytlm:      2000000
Maxacctjobs:     0  Shrfillm:      0  Pbytlm:           0
Maxdetach:       0  BIOlm:       900  JTquota:       8192
Prclm:           0  DIOlm:       900  WSdef:         4096
Prio:            4  ASTlm:       900  WSquo:        16384
Queprio:         0  TQElm:       900  WSextent:     32767
CPU:        (none)  Enqlm:      8192  Pgflquo:    2000000
```

6. Sample applications

The directory `libwww$root:[samples]` contains several simple example programs that illustrate various aspects of the `libwww` library, and the command procedure `samples.com` in this directory may be used to build the examples. These examples provide reasonable coverage of the `libwww` API, and provide a useful basis for the development of more sophisticated applications.