Using Apache With Novell NetWare

This document explains how to install, configure and run Apache 2.0 under Novell NetWare 5.1 and above. If you find any bugs, or wish to contribute in other ways, please use our bug reporting page.[1]

The bug reporting page and dev-httpd mailing list are *not* provided to answer questions about configuration or running Apache. Before you submit a bug report or request, first consult this document, the Frequently Asked Questions[2] page and the other relevant documentation topics. If you still have a question or problem, post it to the novell.devsup.webserver[3] newsgroup, where many Apache users are more than willing to answer new and obscure questions about using Apache on NetWare.

Most of this document assumes that you are installing Apache from a binary distribution. If you want to compile Apache yourself (possibly to help with development, or to track down bugs), see the section on Compiling Apache for NetWare below.

Topics

Requirements

Apache 2.0 is designed to run on NetWare 5.1 and above.

If running on NetWare 5.1 you must install Service Pack 4 or above.

If running on NetWare 6 you must install Service Pack 1 or above.

NetWare service packs are available here[4].

Downloading Apache for NetWare

Information on the latest version of Apache can be found on the Apache web server at http://www.apache.org/. This will list the current release, any more recent alpha or beta-test releases, together with details of mirror web and anonymous ftp sites.

Installing Apache for NetWare

There is no Apache install program for NetWare currently. You will need to compile apache and copy the files over to the server manually. An install program will be posted at a later date.

Follow these steps to install Apache on NetWare from the binary download (assuming you will install to `sys:/apache2`):

- Unzip the binary download file to the root of the `SYS:` volume (may be installed to any volume)
- Edit the `httpd.conf` file setting `ServerRoot` and `ServerName` to reflect your correct server settings

- Add SYS:/APACHE2 to the search path, for example:

```
SEARCH ADD SYS:\APACHE2
```

Follow these steps to install Apache on NetWare manually from your own build source (assuming you will install to sys:/apache):

- Create a directory called Apache2 on a NetWare volume
- Copy APACHE2.NLM, APRLIB.NLM, HTDIGEST.NLM, HTPASSWD.NLM to SYS:/APACHE2
- Create a directory under SYS:/APACHE2 called CONF
- Copy the HTTPD-STD.CONF file to the SYS:/APACHE2/CONF directory and rename to HTTPD.CONF
- Copy the MIME.TYPES and MAGIC files to SYS:/APACHE2/CONF directory
- Copy all files and subdirectories in \HTTPD-2.0\DOCS\ICONS to SYS:/APACHE2/ICONS
- Copy all files and subdirectories in \HTTPD-2.0\DOCS\MANUAL to SYS:/APACHE2/MANUAL
- Copy all files and subdirectories in \HTTPD-2.0\DOCS\ERROR to SYS:/APACHE2/ERROR
- Copy all files and subdirectories in \HTTPD-2.0\DOCS\DOCROOT to SYS:/APACHE2/HTDOCS
- Create the directory SYS:/APACHE2/LOGS on the server
- Create the directory SYS:/APACHE2/CGI-BIN on the server
- Create the directory SYS:/APACHE2/MODULES and copy all nlm modules into the modules directory
- Edit the HTTPD.CONF file searching for all @@Value@@ markers and replacing them with the appropriate setting
- Add SYS:/APACHE2 to the search path, for example:

```
SEARCH ADD SYS:\APACHE2
```

Apache may be installed to other volumes besides the default SYS volume.

Running Apache for NetWare

To start Apache just type apache at the console. This will load apache in the OS address space. If you prefer to load Apache in a protected address space you may specify the address space with the load statement as follows:

```
load address space = apache2 apache2
```

This will load Apache into an address space called apache. Running multiple instances of Apache concurrently on NetWare is possible by loading each instance into its own protected address space.

After starting Apache, it will be listening to port 80 (unless you changed the Listen directive in the configuration files). To connect to the server and access the default page, launch a browser and enter the server's name or address. This should respond with a welcome page, and a link to the Apache manual. If nothing happens or you get an error, look in the error_log file in the logs directory.

Once your basic installation is working, you should configure it properly by editing the files in the conf directory.

To unload Apache running in the OS address space just type the following at the console:

```
unload apache2
```

or

```
apache2 shutdown
```

If apache is running in a protected address space specify the address space in the unload statement:

```
unload address space = apache2 apache2
```

When working with Apache it is important to know how it will find the configuration files. You can specify a configuration file on the command line in two ways:

- `-f` specifies a path to a particular configuration file

```
apache2 -f "vol:/my server/conf/my.conf"
```

```
apache -f test/test.conf
```

In these cases, the proper `ServerRoot` should be set in the configuration file.

If you don't specify a configuration file name with `-f`, Apache will use the file name compiled into the server, usually `conf/httpd.conf`. Invoking Apache with the `-V` switch will display this value labeled as `SERVER_CONFIG_FILE`. Apache will then determine its `ServerRoot` by trying the following, in this order:

- A `ServerRoot` directive via a `-C` switch.
- The `-d` switch on the command line.
- Current working directory
- The server root compiled into the server.

The server root compiled into the server is usually `sys:/apache2`. invoking apache with the `-V` switch will display this value labeled as `HTTPD_ROOT`.

Apache 2.0 for NetWare includes a set of command line directives that can be used to modify or display information about the running instance of the web server. Each of these directives must be preceded by the keyword `APACHE2`:

RESTART
Instructs Apache to terminate all running worker threads as they become idle, reread the configuration file and restart each worker thread based on the new configuration.

VERSION
Displays version information about the currently running instance of Apache.

MODULES
Displays a list of loaded modules both built-in and external.

DIRECTIVES
Displays a list of all available directives.

SETTINGS
Enables or disables the thread status display on the console. When enabled, a status of the number of running threads is displayed along with their status.

SHUTDOWN
Terminates the running instance of the Apache web server.

Configuring Apache for NetWare

Apache is configured by files in the `conf` directory. These are the same as files used to configure the Unix version, but there are a few different directives for Apache on NetWare. See the Apache documentation[5] for all the available directives.

The main differences in Apache for NetWare are:

- Because Apache for NetWare is multithreaded, it does not use a separate process for each request, as Apache does in some Unix implementations. Instead there are only threads running: a parent thread, and a multiple child threads which handle the requests.

  So the "process"-management directives are different:

  `MaxRequestsPerChild` - Like the Unix directive, this controls how many requests a worker thread will serve before exiting. The recommended default, `MaxRequestsPerChild 0`, causes the thread to continue servicing request indefinitely. It is recommended on NetWare, unless there is some specific reason, that this directive always remain set to `0`.

  `StartThreads` - This directive tells the server how many threads it should start initially. The recommended default is `StartThreads 50`.

  `MinSpareThreads` - This directive instructs the server to spawn additional worker threads if the number of idle threads ever falls below this value. The recommended default is `MinSpareThreads 10`.

  `MaxSpareThreads` - This directive instructs the server to begin terminating worker threads if the number of idle threads ever exceeds this value. The recommended default is `MaxSpareThreads 100`.

  `MaxThreads` - This directive limits the total number of work threads to a maximum value. The recommended default is `ThreadsPerChild 250`.

  `ThreadStackSize` - This directive tells the server what size of stack to use for the individual worker thread. The recommended default is `ThreadStackSize 65536`.

- The directives that accept filenames as arguments now must use NetWare filenames instead of Unix ones. However, because Apache uses Unix-style names internally, you must use forward slashes, not backslashes. It is recommended that all rooted file paths begin with a volume name. If omitted, Apache will assume the `SYS:` volume.

- Apache for NetWare has the ability to load modules at runtime, without recompiling the server. If Apache is compiled normally, it will install a number of optional modules in the `\Apache2\modules` directory. To activate these, or other modules, the `LoadModule` directive must be used. For example, to active the status module, use the following (in addition to the status-activating directives in `access.conf`):

  ```
  LoadModule status_module modules/status.nlm
  ```

Information on creating loadable modules[6] is also available.

Additional NetWare specific directives:

- `CGIMapExtension` - This directive maps a CGI file extension to a script interpreter.

Compiling Apache for NetWare

Compiling Apache requires MetroWerks CodeWarrior 6.x or higher to be properly installed. Once Apache has been built, it needs to be installed on a NetWare volume's root directory. The default is the `sys:/Apache2` directory.

Before running the server you must fill out the `conf` directory. Copy the file `HTTPD-STD.CONF` from the distribution `conf` directory and rename it to `HTTPD.CONF`. Edit the `HTTPD.CONF` file searching for all `@@Value@@` markers and replacing them with the appropriate setting. Copy over the `conf/magic` and `conf/mime.types` files as well.

Requirements:

The following development tools are required to build Apache 2.0 for NetWare:

- Metrowerks CodeWarrior 6.0 or higher with the NetWare PDK 3.0[7] or higher.
- NetWare Libraries for C (LibC)[8]
- WinSock 2 Developer Components for NetWare[9]
- To build using either the project file or the make files, requires an AWK utility (awk, gawk or similar). AWK can be downloaded from http://developer.novell.com/ndk/apache.htm. The utility must be found in your windows path and must be named `awk.exe`.
- To build using the makefiles, you will need GNU make version 3.78.1 (GMake) available at http://developer.novell.com/ndk/apache.htm.

Building Apache using the Metrowerks Project Files:

All major pieces of Apache and APR are built using the `ApacheNW.mcp` and `LibAprNW.mcp` project files. This includes modules such as status, info, proxy, etc.

- Set the environment variable `NovellLibC` to the location of the NetWare Libraries for C SDK, for example:

  ```
  Set NovellLibC=c:\novell\ndk\libc
  ```

- Make sure that the path to the CodeWarrior command line tools (`MWCCNLM.exe`, `MWLDNLM.exe`) has been included in the system's `PATH` environment variable.
- Make sure that the path to the AWK utility has been included in the system's `PATH` environment variable.
- Download the source code and unzip to an appropriate directory on your workstation.
- Change directory to `\httpd\srclib\apr\build` and run the batch file `prebuildnw.bat`. The batch file will setup the build environment for building the APR libraries. It will also run 2 AWK scripts that will generate the export files for APR.
- Change directory to `\httpd\srclib\apr` and extract the project file `LIBAPRNW.mcp` from the `LIBAPRNW.mcp.zip` file.
- Open the `LIBAPRNW.mcp` project file in the Metrowerks IDE.

- Select the target "Build Util - Gen URL Delim" and build the target. This target will produce the NLM `GENURI.nlm`
- Copy the file `GENURI.nlm` to the `SYS:` volume of a NetWare server and run using the following command:
  - `SYS:\genuri > sys:\uri_delims.h`
- Copy the file `uri_delims.h` to the directory `\httpd\srclib\apr-util\uri` on the build machine.
- Select the target "APR Debug NLM" or "APR Release NLM" in the IDE and build. This will produce the file `APRLIB.nlm`.
  - [Optional] Select any of the LIB targets to produce a statically linkable libraries.
- Change directory to `\httpd\build` and run the batch file `prebuildnw.bat`. This batch file will setup the build environment for building the `APACHE.nlm`. It will also run several AWK scripts that will generate the export files for APACHE.
- Change directory to `\http` and extract the project file `ApacheNW.mcp` from the `ApacheNW.mcp.zip` file.
- Open the `ApacheNW.mcp` project file in the Metrowerks IDE.
- Select the target "Build Utility - DFTables" and build the target.
- Select the target "Build Util - Gen Test Chars" and build the target.
- Copy the files `GENCHARS.nlm` and `DFTABLES.nlm` to the `SYS:` volume of a NetWare server and run using the following commands:
  - `SYS:\genchars > sys:\test_char.h`
  - `SYS:\dftables > sys:\chartables.c`
- Copy the files `test_char.h` and `chartables.c` to the directory `\httpd\os\netware` on the build machine.
- Select the target "Apache Full Debug" or "Apache Full Release" in the IDE and build. This will produce the file `APACHE2.nlm` along with all of the external module NLMs.

Building Apache using the NetWare makefiles:

- Set the environment variable `NOVELLLIBC` to the location of the NetWare Libraries for C SDK, for example:

  ```
  Set NOVELLLIBC=c:\novell\ndk\libc
  ```

- Set the environment variable `METROWERKS` to the location where you installed the Metrowerks CodeWarrior compiler, for example:

  ```
  Set METROWERKS=C:\Program Files\Metrowerks\CodeWarrior
  ```

  If you installed to the default location `C:\Program Files\Metrowerks\CodeWarrior`, you don't need to set this.
- Set the environment variable `AP_WORK` to the full path of the `\httpd` directory.
- Set the environment variable `APR_WORK` to the full path of the `\httpd\srclib\apr` directory.
- Make sure that the path to the AWK utility and the GNU make utility (`gmake.exe`) have been included in the system's `PATH` environment variable.
- Download the source code and unzip to an appropriate directory on your workstation.

- Change directory to `\httpd\srclib\apr-util\uri` and build `GENURI.nlm` by running "`gmake -f nwgnumakefile`".
- Copy the file `GENURI.nlm` to the `SYS:` volume of a NetWare server and run using the following command:

  ```
  SYS:\genuri > sys:\uri_delims.h
  ```

- Copy the file `uri_delims.h` to the directory `\httpd\srclib\apr-util\uri` on the build machine.
- Change directory to `\httpd\srclib\apr` and build APR by running "`gmake -f nwgnumakefile`"
- Change directory to `\httpd\srclib\pcre` and build `DFTABLES.nlm` by running "`gmake -f nwgnumakefile`"
- Change directory to `\httpd\server` and build `GENCHARS.nlm` by running "`gmake -f nwgnumakefile`"
- Copy the files `GENCHARS.nlm` and `DFTABLES.nlm` from their respective directories to the `SYS:` volume of a NetWare server and run them using the following commands:

  ```
  SYS:\genchars > sys:\test_char.h
  SYS:\dftables > sys:\chartables.c
  ```

- Copy the files `test_char.h` and `chartables.c` to the directory `\httpd\os\netware` on the build machine.
- Change directory to `\httpd` and build Apache by running "`gmake -f nwgnumakefile`". You can create a distribution directory by adding an install parameter to the command, for example:

  ```
  gmake -f nwgnumakefile install
  ```

Additional make options

- `gmake -f nwgnumakefile`
  Builds release versions of all of the binaries and copies them to a `\release` destination directory.
- `gmake -f nwgnumakefile DEBUG=1`
  Builds debug versions of all of the binaries and copies them to a `\debug` destination directory.
- `gmake -f nwgnumakefile install`
  Creates a complete Apache distribution with binaries, docs and additional support files in a `\dist\Apache2` directory.
- `gmake -f nwgnumakefile installdev`
  Same as install but also creates a `\lib` and `\include` directory in the destination directory and copies headers and import files.
- `gmake -f nwgnumakefile clean`
  Cleans all object files and binaries from the `\release` or `\debug` build areas depending on whether `DEBUG` has been defined.
- `gmake -f nwgnumakefile clobber_all`
  Same as clean and also deletes the distribution directory if it exists.

URI References

[1] http://httpd.apache.org/bug_report.html

[2] http://httpd.apache.org/docs-2.1/faq/
[3] news://developer-forums.novell.com/novell.devsup.webserver
[4] http://support.novell.com/misc/patlst.htm#nw
[5] http://httpd.apache.org/docs-2.1/
[6] http://httpd.apache.org/docs-2.1/mod/mod_so.html#creating
[7] http://developer.novell.com/ndk/cwpdk.htm
[8] http://developer.novell.com/ndk/libc.htm
[9] http://developer.novell.com/ndk/ws2comp.htm