

Web Server Manual

Mike Burns (netgeek@speakeasy.net)
Greg Pettyjohn (gregp@ccs.neu.edu)
Jay McCarthy (jay.mccarthy@gmail.com)

November 20, 2006

Copyright notice

Copyright ©1996-2006 PLT

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Library General Public License, Version 2 published by the Free Software Foundation. A copy of the license is included in the appendix entitled "License."

Contents

1. Quick Start

To just use a servlet without caring about administrative or programming details:

1.1 Quick Start on Unix

1. If you have root access, just run `plt/bin/web-server`; otherwise, run `plt/bin/web-server -p 8080`.
2. Example servlets are located in `collects/web-server/default-web-root/servlets/examples/`.
3. To run the example servlet **add.ss**, use the URL <http://localhost/servlets/examples/add.ss>, or <http://localhost:8080/servlets/examples/add.ss> if the server was started with `-p 8080`.

Part I

Administration

2. Directory Structure

By default, the configuration tool (see section 3.2) organizes files containing the Web server's configurations, documents, and servlets in one directory tree per virtual host. This tool can be used to manage all trees for the Web server. A default tree exists in `collection/web-server/`, which looks like:

```
configuration-table
default-web-root
  conf
    servlet-error.html
    forbidden.html
    servlet-refresh.html
    passwords-refresh.html
    not-found.html
    protocol-error.html
  htdocs
    Defaults
      index.html
      documentation
    ...
  log
  passwords
  servlets
    configure.ss
my-other-host
  conf ...
  htdocs ...
  log
  passwords
  servlets
still-another-host ...
```

Files may be relocated or shared between hosts by editing the details for that host using the configuration tool. For more on virtual hosts, see section 3.1.

3. Configuration

3.1 Named Virtual Hosts

Many Web sites can be served from one Web server process through the use of virtual hosting. When a client (Web browser) sends a request to the server, the domain name is embedded in the request. The Web server can then service the request in different ways based on that requested domain name.

Zero or more virtual hosts may be defined, in addition to the default host. If a request is sent to the server for a host which is not in the virtual host table, then the default host is used. For example, if the client requests <http://www.plt-scheme.org/>, the server can send back information in the directory `/var/www/plt/www.plt-scheme.org/`; if the client requests, from the save server, <http://www.drscheme.org/>, the server can send back information from the directory `/home/mburns/www.drscheme.org/`.

To configure virtual hosting using the configuration tool, see section 3.2.1. To configure virtual hosting by editing the file directly, see `virtual-host-table`.

3.2 The Configuration Tool

The configuration tool is a normal servlet, run by the Web server, that can manage the configuration files. This is provided as an alternative to editing the file directly.

For security reasons, the configuration servlet is only accessible from the local machine. Thus, one gets to the servlet via <http://localhost/servlets/configure.ss>.

Once there, select the filename for the configuration file (the default should work). As part of the basic configuration, you can change the port to listen on, the maximum number of waiting connections, and the initial timeout. Individual hosts can be added and edited below the basic configuration.

3.2.1 Managing Virtual Hosts

Adding a virtual host is as simple as selecting “Add Host.” The page will reload with the information for the new host; simply change the name to be the domain name for the new host, and change the directory to be the correct directory for the virtual host. The Web server must be restarted for these settings to take effect.

In addition to the virtual hosts’s domain and Web root, the details of all hosts (including the default host) can be changed by selecting “Edit Minor Details.” From there, the log file, static files directory, password file, timeout values, and message files can all be modified. For details, see section 3.3.

3.3 Configuration File Syntax

Instead of using the configuration tool, as described in section 3.2, the configuration file can be edited “by hand”. The provided, default configuration file is `collects/web-server/configuration-table`. The file has the following syntax.

```
((port natural-number)
 ((max-waiting natural-number)
 ((initial-connection-timeout number)
 ((default-host-table host-table)
 ((virtual-host-table (string host-table) ...)))
```

`port` : `natural-number` Usually 80, indicating which port to listen on by default.

`max-waiting` : `natural-number` Usually 40, indicating the maximum number of clients that can wait for a TCP/IP connection to the Web server. This limit usually does not cause problems because the Web server quickly accepts connections as requested and there is no limit on the number of simultaneous connections. On some platforms, choosing a large number may consume too many resources.

`initial-connection-timeout` : `natural-number` Usually 30. After a client connects to the Web server, the server only waits for an HTTP request for this many seconds before closing the connection. Browsers tend to send requests immediately, so a small number is best. This number should be kept small to prevent denial-of-service attacks.

`default-host-table` : `host-table` Described in section 3.3.1.

`virtual-host-table` : `(listof (cons string host-table))` A list of `(string host-table)`, where the `host-table` is defined in section 3.3.1 and the `string` represents the named virtual host's domain name.

For example, let the virtual host table be

```
(virtual-host-table
 ("www.plt-scheme.org"
 (host-table
 (default-indices "index.shtml" "index.html")
 (log-format parenthesized-default)
 (messages
 (servlet-message "servlet-error.html")
 (authentication-message "forbidden.html")
 (servlets-refreshed "servlet-refresh.html")
 (passwords-refreshed "passwords-refresh.html")
 (file-not-found-message "not-found.html")
 (protocol-message "protocol-error.html")
 (collect-garbage "collect-garbage.html")))
 (timeouts
 (default-servlet-timeout 120)
 (password-connection-timeout 300)
 (servlet-connection-timeout 86400)
 (file-per-byte-connection-timeout 1/20)
 (file-base-connection-timeout 30))
 (paths
 (configuration-root "errors")
 (host-root "/var/www/www.plt-scheme.org")
 (log-file-path "/var/log/www.plt-scheme.org")
 (file-root "htdocs")
 (servlet-root ".")
 (mime-types "mime.types")
 (password-authentication "passwd"))))))
```

Requesting the URL <http://localhost/> will use the `default-host-table`, but requesting the URL <http://www.plt-scheme.org/>, assuming DNS is set up correctly, will return the first of

1. `/var/www/www.plt-scheme.org/index.shtml`
2. `/var/www/www.plt-scheme.org/index.html`

For port-based virtual hosting, run separate instances of the Web server.

3.3.1 Host Table Syntax

The host table is a part of the configuration file with the following syntax

```
(host-table
 (default-indices string ...)
 (log-format parenthesized-default)
 (messages
 (servlet-message string)
 (authentication-message string)
 (servlets-refreshed string)
 (passwords-refreshed string)
 (file-not-found-message string)
 (protocol-message string)
 (collect-garbage string))
 (timeouts
 (default-servlet-timeout number)
 (password-connection-timeout number)
 (servlet-connection-timeout number)
 (file-per-byte-connection-timeout number)
 (file-base-connection-timeout number))
 (paths
 (configuration-root string)
 (host-root string)
 (log-file-path string)
 (file-root string)
 (servlet-root string)
 (password-authentication string)))
```

`default-indices` : (listof *string*) This is a list of index files for all directories. When a directory is specified as the URL, such as `http://www.plt-scheme.org/`, the Web server searches for a file to display, choosing from this list in the specified order. A good default is

```
(default-indices "index.html" "index.htm")
```

For example, if `default-indices` is set to

```
(default-indices "index.ss" "index.html" "index.htm" "last-resort")
```

and the user requests `http://www.plt-scheme.org/`, the Web server will attempt to serve the following URLs, in order, until a file is found:

1. `http://www.plt-scheme.org/index.ss`
2. `http://www.plt-scheme.org/index.html`
3. `http://www.plt-scheme.org/index.htm`
4. `http://www.plt-scheme.org/last-resort`

`log-format` : *symbol* The format to use for the log file. Currently only the symbol `parenthesized-default` is supported.

3.3.1.1 MESSAGES

`servlet-message` : `string` A filename, usually `"servlet-error.html"`, of the page to display when there is an error with the servlet. This file is located relative to the `configuration-root` directory.

`authentication-message` : `string` A filename, usually `"forbidden.html"`, of the page to display when there user fails authentication. This file is located relative to the `configuration-root` directory.

`servlets-refreshed` : `string` A filename, usually `"servlet-refresh.html"`, of the page to display when the servlets are refreshed. This file is located relative to the `configuration-root` directory.

`passwords-refreshed` : `string` A filename, usually `"passwords-refresh.html"`, of the page to display when the password file is refreshed. This file is relative to the `configuration-root` directory.

`file-not-found-message` : `string` A filename, usually `"not-found.html"`, of the page to display when the requested file was not found (HTTP error number 404). This file is located relative to the `configuration-root` directory.

`protocol-message` : `string` A filename, usually `"protocol-error.html"`, of the page to display when the Web browser violates the HTTP standard. This should never appear when interacting with the server through a correct Web browser. The server may close the connection instead of sending this error, depending on the severity of the problem.

`collect-garbage` : `string` A filename, usually `"collect-garbage.html"`, of the page to display when the garbage collected is run. This file is relative to the `configuration-root` directory.

3.3.1.2 TIMEOUTS

`default-servlet-timeout` : `number` This value, usually 120, determines how long, in seconds, a servlet can wait in between requests before shutting down, unless the servlet calls `adjust-timeout!` to alter this value. Large values consume more memory, while smaller values annoy end users who must restart servlets.

`password-connection-timeout` : `number` Usually 300, this is the number of seconds a user has to enter a username and password.

`servlet-connection-timeout` : `number` Usually 86400 (one day), this is the number of seconds a servlet has to respond to a single request. Some servlets may perform substantial computation on a loaded server before generating a page. On the other hand, most users will not wait very long for a response.

`file-per-byte-connection-timeout` : `number` Usually 1/20. Let *bytes* be the bytes in a file. The total number of seconds the server has to send a file is `file-base-connection-timeout + file-per-byte-connection-timeout * bytes`

`file-base-connection-timeout` : `number` Usually 30. Let *bytes* be the bytes in a file. The total number of seconds the server has to send a file is `file-base-connection-timeout + file-per-byte-connection-timeout * bytes`

3.3.1.3 PATHS

`configuration-root` : `string` The directory under which the files in section 3.3.1.1 may be found. Usually `"conf"`.

`host-root` : `string` The root directory for Web files. Under this directory is usually the `servlet-root`, `configuration-root`, `log-file-path`, `file-root`, and `password-authentication`. Usually `"default-web-root"`.

`log-file-path` : `string` The name of the log file. Usually `"log"`.

`file-root` : string The directory from which static (HTML) files are served. Usually "htdocs".

`servlet-root` : string The directory from which dynamic (servlet) files are run and served. Causes odd behavior when set to anything besides ". ". The servlets directory is always `servlets/`, under the `host-root`.

`mime-types` : string The file from which MIME types are read. For information on its format, see Apache's documentation on the subject, http://httpd.apache.org/docs/1.3/mod/mod_mime.html#typesconfig. Usually "mime.types".

`password-authentication` : string The file from which passwords are read. For information on its format, see section 4. Usually "passwords".

4. Passwords

The PLT Web server provides *basic* authentication following RFC 2617. These passwords are stored in **passwords**. The Web server caches passwords for performance reasons. Requesting the URL <http://localhost/conf/refresh-passwords> reloads the password file. The format is:

```
'((realm path-regexp (name password) ...) ...)
```

realm : *string* A unique identifier for this set of passwords.

path-regexp : *string* A regular expression applied to the resource. If it matches, then this set of passwords is used for this resource.

name : *symbol* The login name for a user. Note that case-sensitivity depends on the current settings in MzScheme.

password : *string* The password, *unencrypted*, for the user.

For example, to hide anything in the `/secret/` directory under the `host-root`, but allow access to `bubba` with the password `bbq` and `Billy` with the password `BoB`, use

```
'(("secret stuff" "/secret(/.*)?" (bubba "bbq") (|Billy| "BoB")))
```

5. Starting the Server

The Web server collection provides two programs for starting the Web server. Both accept the same options.

web-server Servlets can use the full MrEd functionality.

web-server-text Servlets can only use MzScheme functionality. This server cannot load servlets written using DrScheme's graphical XML boxes, uses less memory, and works with the MzScheme-only distribution of PLT.

The command line

```
web-server[-text] [-p <port>] [-f <configuration-table-file>] [-a <ip-address>]
```

starts the server on port 80 or *port* and uses the configuration options from the **configuration-table** file of the **web-server** collection or from the specified *configuration-table-file*. If *ip-address* is provided, the server binds to *ip-address*.

6. Monitoring the Server

When launched via

```
web-server-monitor [-p <port>]
                  [-f <frequency>]
                  [-t <timeout>]
                  <alert-email> <host-name>
```

the **web-server-monitor** polls any Web server running on host *host-name* at port *port* (or port 80) every *frequency* seconds (or 1 hour). If the server does not respond to a HEAD HTTP request for the homepage within *timeout* (or 75) seconds or sends an error response, the monitor will notify *alert-email* of the problem.

Part II

Programming

7. Before You Begin

7.1 Reloading the Cache

For efficiency, servlets are cached on their initial invocation. When editing a servlet, before changes can be observed, it is necessary to clear the cache by loading the URL <http://localhost/conf/refresh-servlets>.

7.2 Directories

Within the Web server's root directory, two important directories exist. Note that these names may change. Administrators should see section 2 and section 3.3.1.3.

htdocs/ Under this directory exists static files which are just presented to the Web browser with an appropriate MIME type. See section 8 for the details.

servlets/ Under this directory exists dynamic files which are processed by the Web server as servlets. See section 9 for the details.

8. Static Content

The Web server serves content statically or dynamically, depending on the request URL.

On static requests (those that do not match the Web server's servlet filter), the Web server serves files out of the content directory determined from `file-root` for the specified virtual host.

The Web server guarantees that for such static responses, it will serve only those files accessible in subdirectories of the content directory. In the absence of filesystem links, this means that only files which live in the content directory will be made available to the outside world. The server ignores the relative path specifiers `..` and `.` in URLs.

9. Writing Servlets

When a request URL matches the servlet filter, the Web server generates its response dynamically, (see section 7.2 for more explanation, and section 2 and section 3.3.1.3 for administrative details). Instead of serving the files in this directory verbatim, the server evaluates the file as a scheme program to produce output. Servlets are (by default) loaded in a case-sensitive manner. (Search in help-desk for `read-case-sensitive`.)

The path part of the URL supplies the file path to the servlet relative to the `servlets` directory. However, paths may also contain extra path components that servlets may use as additional input. For example all of the following URLs refer to the same servlet:

- `http://www.plt-scheme.org/servlets/my-servlet`
- `http://www.plt-scheme.org/servlets/my-servlet/extra`
- `http://www.plt-scheme.org/servlets/my-servlet/extra/directories`

The server supports two forms of servlets. For useful procedures to handle Web data, see section 10.

9.1 Module-Based Servlets

A module-based servlet is a module that provides three values: an `interface-version`, a `timeout`, and a `start` procedure.

```
(module a-module-servlet mzscheme
  (provide interface-version timeout start)

  (define interface-version 'v1)

  (define timeout +inf.0)

  ; start : request → response
  (define (start initial-request)
    `(html (head (title "A Test Page"))
           (body ([bgcolor "white"])
                 (p "This is a simple module servlet."))))))
```

`interface-version`: symbol

The `interface-version` is a symbol indicating how the server should interact with the servlet. The only supported value is `'v1` at this time.

```
timeout : number
```

The `timeout` is the number of seconds the server will allow the servlet to run before shutting it down. Large values consume more memory, while smaller values annoy users by forcing them to restart their session. The value can be adjusted dynamically by calling the `adjust-timeout!` procedure. For more information, see section 3.3.1.2 and section ??.

start

```
request -> response
(define (start request) ...)
```

The `start` function consumes a `request` and produces a `response`. Each time a client visits the URL associated with the beginning of the servlet, the server calls the `start` function with the `request` sent from the browser. The server then sends the `response` produced by the `start` function back to the browser.

9.2 Unit-Based Servlets

A unit-based servlet is a `unit/sig` that imports the `servlet^` signature and exports nothing. (See the manual for `unit/sig` and `signatures`.) To construct a signed unit with the appropriate imports, the servlet must require the two modules providing `unit/sigs` and the `servlet` signature:

```
(require (lib "unitsig.ss")
         (lib "servlet-sig.ss" "web-server"))
(unit/sig ()
 (import servlet^)
 ;; ...insert servlet code here...
)
```

Evaluating

```
(require (lib "servlet-sig.ss" "web-server"))
```

loads the `servlet^` signature which contains the import `initial-request` of type `request`.

10. Servlet Library

To ease the development of interactive servlets, the **web-server** collection also provides the following data types and procedures:

10.1 Data Definitions

10.1.1 Environment

An environment is a `(listof (cons symbol string))`. That is, an environment is a list of improper cons-pairs of a symbol and a string.

A byte-environment is a `(listof (cons symbol bytes))`. That is, a byte-environment is a list of improper cons-pairs of a symbol and a bytes.

10.1.2 Request

A request is

```
%(make-request symbol url environment environment string string)
(define-struct request (method uri headers bindings bindings/raw host-ip client-ip post-
```

`method` One of

- `'get`
- `'post`

`uri` URL, see the **net** collection in help-desk for details.

`headers` An environment containing optional HTTP headers for this request.

`bindings` An environment containing optional name-value pairs from either the form submitted or the query part of the URL.

`bindings/raw` Either a string or a byte-environment. This is primarily used for file uploads.

`host-ip` A string representing what IP address of the server the request came to.

`client-ip` A string representing what IP address is associated with the client.

`post-data/raw` Either false or a byte-string representing the full POST data.

The bindings contain the information the user supplied when filling out a Web form and hence are usually the most useful part. This makes the following idiom common. For a full example, see Figure ??.

```
(extract-bindings/single
  'sym
```