Apache's Handler Use

This document describes the use of Apache's Handlers.

Topics

What is a Handler

Related Modules	Related Directives
mod_actions	Action
mod_asis	AddHandler
mod_cgi	RemoveHandler
mod_imap	SetHandler
mod_info	
mod_mime	
mod_negotiation	
mod_status	

A "handler" is an internal Apache representation of the action to be performed when a file is called. Generally, files have implicit handlers, based on the file type. Normally, all files are simply served by the server, but certain file types are "handled" separately.

Apache 1.1 adds the ability to use handlers explicitly. Based on either filename extensions or on location, handlers can be specified without relation to file type. This is advantageous both because it is a more elegant solution, and because it also allows for both a type **and** a handler to be associated with a file. (See also Files with Multiple Extensions[1].)

Handlers can either be built into the server or included in a module, or they can be added with the `Action` directive. The built-in handlers in the standard distribution are as follows:

- **default-handler**: Send the file using the `default_handler()`, which is the handler used by default to handle static content. (core)
- **send-as-is**: Send file with HTTP headers as is. (`mod_asis`)
- **cgi-script**: Treat the file as a CGI script. (`mod_cgi`)
- **imap-file**: Parse as an imagemap rule file. (`mod_imap`)
- **server-info**: Get the server's configuration information. (`mod_info`)
- **server-status**: Get the server's status report. (`mod_status`)
- **type-map**: Parse as a type map file for content negotiation. (`mod_negotiation`)

Examples

Modifying static content using a CGI script

The following directives will cause requests for files with the `html` extension to trigger the launch of the `footer.pl` CGI script.

```
Action add-footer /cgi-bin/footer.pl
```

```
AddHandler add-footer .html
```

Then the CGI script is responsible for sending the originally requested document (pointed to by the `PATH_TRANSLATED` environment variable) and making whatever modifications or additions are desired.

Files with HTTP headers

The following directives will enable the `send-as-is` handler, which is used for files which contain their own HTTP headers. All files in the `/web/htdocs/asis/` directory will be processed by the `send-as-is` handler, regardless of their filename extensions.

```
<Directory /web/htdocs/asis>
SetHandler send-as-is
</Directory>
```

Programmer's Note

In order to implement the handler features, an addition has been made to the Apache API [2] that you may wish to make use of. Specifically, a new record has been added to the `request_rec` structure:

```
char *handler
```

If you wish to have your module engage a handler, you need only to set `r->handler` to the name of the handler at any time prior to the `invoke_handler` stage of the request. Handlers are implemented as they were before, albeit using the handler name instead of a content type. While it is not necessary, the naming convention for handlers is to use a dash-separated word, with no slashes, so as to not invade the media type name-space.

URI References

[1] http://httpd.apache.org/docs-2.1/mod/mod_mime.html#multipleext
[2] http://httpd.apache.org/docs-2.1/developer/API.html