Hypertext Transfer Protocol (HTTP) and the Apache HTTP Server

April 11, 2006

Harvard University
Division of Continuing Education
Extension School

Course Web Site: http://cscie12.dce.harvard.edu/
Copyright 1998-2006 David P. Heitmeyer
My email: david_heitmeyer@harvard.edu
Course staff email: cscie12@fas.harvard.edu

TCP/IP: Transmission Control Protocol/Internet Protocol

The TCP/IP Internet Protocol Suite

IP (Internet Protocol): provides basic communication rules
TCP (Transmission Control Protocol): provides additional facilities

IP Addresses

The name www.fas.harvard.edu resolves to the IP address of 140.247.34.66. You can use the tool host or dig to lookup the IP to name or name to IP number.

http://www.npr.org/
Hostnames

cscl2.dce.harvard.edu

Numbers

Internet Domain Survey reports 394,991,609 hosts in the Domain Name Service (Source: Internet Software Consortium (http://www.isc.org)).

Application Layer

- HTTP (default port 80)
- FTP (port 21)
- SMTP (port 25)
- telnet (port 23)
- ssh (port 22)

HyperText Transfer Protocol

Specifies the grammar of a conversation between an HTTP-client (Web Browser) and an HTTP-server (Web Server) is to take place.

GET /
HTTP

Example Request and Loading of a Web Page

As an example, an XHTML page with 7 images and an external CSS file and an external Javascript file, the client would make 10 separate requests (1 request for the XHTML resource, 1 request for each of the seven images, 1 request for the CSS, and 1 request for the JS).

HTTP Request for "example.html" HTTP Response with "example.html" content
HTTP Request for CSS document HTTP Response for CSS document
HTTP Request for Javascript document HTTP Response for Javascript document
HTTP Request for image 1 HTTP Response for image 1
HTTP Request for image 2 HTTP Response for image 2
HTTP Request for image 3 HTTP Response for image 3
HTTP Request for image 4 HTTP Response for image 4
HTTP Request for image 5 HTTP Response for image 5
HTTP Request for image 6 HTTP Response for image 6
HTTP Request for image 7 HTTP Response for image 7

Render page

HTTP is Stateless

Each requested resource is a separate, independent, request to the server — it is a stateless protocol.

HTTP Versions

W3C and Internet Engineering Task Force (IETF) oversees the Hypertext Transfer Protocol.

- HTTP 1.0 (1996)
- HTTP 1.1 (1999)
- Extensions to HTTP
  - WebDAV

An HTTP Conversation

- Client Request
  - METHOD Request HTTP Version
  - Request Headers
  - Request Body
- Server Response
  - Status Line
  - Server Generated Headers
  - Data
HTTP 1.1 Methods

- GET
- POST
- HEAD
- PUT
- DELETE
- TRACEx
- OPTIONS

HTTP Response Codes

<table>
<thead>
<tr>
<th>Code Range</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>100's</td>
<td>Informational</td>
</tr>
<tr>
<td>200's</td>
<td>Success</td>
</tr>
<tr>
<td>300's</td>
<td>Redirected</td>
</tr>
<tr>
<td>400's</td>
<td>Request Incomplete</td>
</tr>
<tr>
<td>500's</td>
<td>Server Error</td>
</tr>
</tbody>
</table>

HTTP 1.1 status codes

Common ones:
- 200 OK
- 301 Moved permanently
- 302 Moved temporarily
- 403 Forbidden
- 500 Internal server error

The complete list:
- 100 Continue
- 101 Switching protocols
- 200 OK
- 201 Created
- 202 Accepted
- 300 Multiple choices
- 301 Moved permanently
- 302 Moved temporarily
- 303 See other
- 304 Not modified
- 400 Bad request
- 401 Unauthorized
- 402 Payment required
- 403 Forbidden
- 404 Not found
- 405 Method not allowed
- 406 Not acceptable
- 410 Gone
- 411 Length required
- 412 Precondition failed
- 413 Request entity too large
- 414 Request-URI too long
- 415 Unsupported media type
- 416 Requested range not satisfiable
- 428 Precondition required
- 429 Too many requests
- 500 Internal server error
- 501 Not implemented
- 502 Bad gateway
- 503 Service unavailable
- 504 Gateway timeout
- 505 HTTP version not supported

Common Request Headers

- Accept
- Accept-Language
- Authorization
- Cookie
- Host
- If-Modified-Since
- Referer
- User-Agent
### Sample Request Header Values

View some of the Headers your browser is sending to the server

**Mozilla Firefox**

- `HTTP_ACCEPT` MIME types:
  - text/xml, application/xml, application/xhtml+xml, text/html, image/png, */*; q=0.8
  - image/jpg, image/gif, */*; q=0.5
- `HTTP_ACCEPT_CHARSET`: `ISO-8859-1, utf-8, q=0.7, *; q=0.7`
- `HTTP_ACCEPT_ENCODING`: gzip, deflate
- `HTTP_ACCEPT_LANGUAGE`: en-us, en, fr, ar
- `HTTP_CONNECTION`: keep-alive
- `HTTP_COOKIE`: PINAuthToken=jswYP9KEk5xgfx%2Bf3GPsr8L3AGvyDM1S1s2NlWDCSgq0bshwsz7K%2F%E0Kka%2FS%8H
- `HTTP_HOST`: cscie12.dce.harvard.edu
- `HTTP_USER_AGENT`: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1) Gecko/20060111 Firefox/1.5.0.1

**Opera 8**

- `HTTP_ACCEPT` MIME types:
  - text/html, application/xml;q=0.9, application/xhtml+xml, image/png, image/jpeg, image/gif, */*; q=0.1
- `HTTP_ACCEPT_CHARSET`: windows-1252, utf-8, utf-16, iso-8859-1; q=0.6, *; q=0.1
- `HTTP_ACCEPT_ENCODING`: gzip, deflate, identity, *; q=0
- `HTTP_ACCEPT_LANGUAGE`: en
- `HTTP_CONNECTION`: Keep-Alive
- `HTTP_HOST`: cscie12.dce.harvard.edu
- `HTTP_USER_AGENT`: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; en-US; rv:1.8.1.0) Gecko/20060111 Firefox/1.5.0.1

**Internet Explorer 6**

- `HTTP_ACCEPT` MIME types:
  - image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/vnd.ms-word, */*
- `HTTP_ACCEPT_ENCODING`: gzip, deflate
- `HTTP_ACCEPT_LANGUAGE`: en-us

---

### Experimenting with HTTP

#### telnet

- `lwp-request`

  **Documentation:**
  - `minerva man lwp-request`
  - `minerva lwp-request -H`

  **An example:**
  - `minerva lwp-request -o head http://www.harvard.edu/

#### Firefox Extension - Live HTTP Headers

- ` HTTP Method: HEAD

  ![HTTP Headers](http://localhost:8080/cocoon/projects/cscie12/slides/20060411/handout.html)

---

### Viewing HTTP Request and Response Headers

- `HTTP_METHOD: HEAD`

  - `http://cscie1.dce.harvard.edu/http/raspberry.gif`
  - `http://cscie12.dce.harvard.edu` is 140.247.197.240
  - `HTTP/1.1 200 OK`
  - `HTTP/1.1 200 OK
  - `Server: Apache/2.0.51 (Debian)
  - `Last-Modified: Wed, 06 Apr 2005 19:30:42 GMT
  - `ETag: "461fb8-348c-a0f67c80"
  - `Accept-Ranges: bytes`
  - `Connection: close
  - `Transfer-Encoding: chunked
  - `Content-Type: image/gif
  - `Connection closed by foreign host.`
Sometimes a file is not there...

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Trying /http/blueberry.gif HTTP/1.1
HTTP/1.1 404 Not Found
Date: Tue, 11 Apr 2006 20:24:27 GMT
Server: Apache/2.0.51 (Fedora)
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Problem: "Infinite" domain names; finite IP addresses.

Solution: "Virtual Hosts"

Example: cscie12.dce.harvard.edu, cscie153.dce.harvard.edu, cscisl.dce.harvard.edu and minerva.dce.harvard.edu resolve to the same IP address (140.247.197.240).

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Trying /http/private/ HTTP/1.1
HTTP/1.1 401 Authorization Required
Date: Tue, 11 Apr 2006 20:33:55 GMT
Server: Apache/2.0.51 (Fedora)
WWW-Authenticate: Basic realm="Private Area"
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

HTTP Authentication/Authorization

The username:password is sent MIME BASE 64 encoded (not encrypted). The username:password is guest:knockknock.

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Trying /http/private/ HTTP/1.1
Authorization: BASIC Z3Vlc3Q6a25vY2trbm9jaw==
HTTP/1.1 200 OK
Date: Tue, 11 Apr 2006 20:35:56 GMT
Server: Apache/2.0.51 (Fedora)
Accept-Ranges: bytes
Connection: close
Content-Type: text/html; charset=utf-8
```

HTTP: Authentication/Authorization

The username:password is sent MIME BASE 64 encoded (not encrypted). The username:password is guest:knockknock.
HTTP Redirect

minerva% telnet www.fas.harvard.edu 80
Trying 140.247.197.240...
Connected to www.fas.harvard.edu.
Server: Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6g mod_perl/1.24
Host: www.fas.harvard.edu
HTTP/1.1 301 Moved Permanently
Date: Wed, 06 Apr 2005 20:11:43 GMT
Server: Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6g mod_perl/1.24
Location: http://www.fas.harvard.edu/home/
Content-Type: text/html; charset=iso-8859-1

More information about MIME Types is available.

Questions:
- How does the server know the media type?
- How does the client know the media type?
- How does the client know "what to do with" the file?

Content Negotiation

Resource can be in multiple formats and languages. Client preferences can determine which resource is returned.

Content Negotiation Resources
- ApacheWeek: Content Negotiation Explained
  http://www.apacheweek.com/features/negotiation
- Apache: Content Negotiation
  http://httpd.apache.org/docs/content-negotiation.html

HTTP Transaction:

minerva% telnet 140.247.34.66 80
Trying 140.247.34.66...
Connected to 140.247.34.66.
Escape character is '\]'.
HEAD /http/raspberry HTTP/1.1
Host: cscie12.dce.harvard.edu
HTTP/1.1 200 OK
Date: Tue, 11 Apr 2006 20:39:20 GMT
Server: Apache/2.0.51 (Fedora)
Content-Location: raspberry.png
Vary: negotiate,accept
TCN: choice
Last-Modified: Wed, 06 Apr 2005 15:30:42 GMT
Accept-Ranges: bytes
Connection: close
ETag: "461fba-22c3-a0f67c80;4edcb400"
Accept: gzip, deflate
Content-Length: 8899
Connection closed by foreign host.
Content Negotiation: MIME Types

Client specifies MIME Types it accepts through HTTP "Accept" header.

A file listing:

```
minerva% ls -l lang*
-rw-r--r--  1 e12  e12  191 Apr  6 15:30 lang.html.de
-rw-r--r--  1 e12  e12  193 Apr  6 15:30 lang.html.en
-rw-r--r--  1 e12  e12  191 Apr  6 15:30 lang.html.fr
```

HTTP Transaction:

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is '^]'.
HEAD /http/lang HTTP/1.1
Host: cscie12.dce.harvard.edu
Connection: close

HTTP/1.1 200 OK
Date: Tue, 11 Apr 2006 20:41:02 GMT
Server: Apache/2.0.51 (Fedora)
Content-Location: lang.html.en
Vary: negotiate, accept-language
TCN: choice
Accept-Ranges: bytes
Content-Length: 193
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Language: en

Connection closed by foreign host.
```

Content Negotiation: Language

Permissions must be set to `rwxr-xr-x` on directories.

Why?

Content Negotiation Gotcha

```
minerva% ls -l lang*
-rw-r--r--  1 e12  e12  191 Apr  6 15:30 lang.html.de
-rw-r--r--  1 e12  e12  193 Apr  6 15:30 lang.html.en
-rw-r--r--  1 e12  e12  191 Apr  6 15:30 lang.html.fr
```

HTTP Transaction:

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is '^]'.
HEAD /http/lang HTTP/1.1
Host: cscie12.dce.harvard.edu
Connection: close
Accept-Language: de

HTTP/1.1 200 OK
Date: Tue, 11 Apr 2006 20:44:16 GMT
Server: Apache/2.0.51 (Fedora)
Content-Location: lang.html.de
Vary: negotiate, accept-language
TCN: choice
Accept-Ranges: bytes
Content-Length: 191
Connection: close
Content-Type: text/html; charset=UTF-8
Content-Language: de

Connection closed by foreign host.
```
HTTP: Moved

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is "^]".
```

HTTP/1.1 301 Moved Permanently
Date: Tue, 11 Apr 2006 20:47:19 GMT
Server: Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6g mod_perl/1.24
Location: http://www.courses.fas.harvard.edu/~cscie12/
Content-Type: text/html; charset=iso-8859-1

```
HTTP: trailing 'slash' for directories
```

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is "^]".
```

HTTP/1.1 301 Moved Permanently
Date: Tue, 11 Apr 2006 20:48:42 GMT
Server: Apache/2.0.51 (Fedora)
Location: http://cscie12.dce.harvard.edu/images/
Content-Type: text/html

```
HTTP 1.1 Improvements over HTTP 1.0
```

- New request methods
- Persistent Connections
- Chunked Encoding
- Byte Range Operations
- Content Negotiation
- Digest Authentication
- Caching

```
Connection: keep-alive
```

```
Connection: close
HTTP/1.1 200 OK
```

```
Connection closed by foreign host.
```
Caching Related Headers

Local cache and Proxy-server cache

- If-Modified-Since
- Age
- Expires
- Last-Modified
- Cache-Control
- ETag

HTTP Example

```
minerva% telnet www.apache.org 80
Trying 192.87.106.226...
Connected to www.apache.org.
Escape character is '^]'.
HEAD /images/asf_logo.gif HTTP/1.1
Host: www.apache.org
Connection: close

HTTP/1.1 200 OK
Date: Tue, 11 Apr 2006 20:52:43 GMT
Server: Apache/2.2.0 (Unix)
Last-Modified: Sun, 20 May 2001 02:19:30 GMT
ETag: "c558a6-1c6f-bf85080"
Accept-Ranges: bytes
Content-Length: 7279
Cache-Control: max-age=86400
Expires: Wed, 12 Apr 2006 20:52:43 GMT
Connection: close
Content-Type: image/gif
```

```
minerva% telnet www.harvard.edu 80
Trying 128.103.60.55...
Connected to zooey.harvard.edu.
Escape character is '^]'.
GET /images/global/banner.gif HTTP/1.1
Host: www.harvard.edu
Keep-Alive: 300
Connection: keep-alive
If-Modified-Since: Thu, 12 Sep 2002 21:51:13 GMT

HTTP/1.1 304 Not Modified
Date: Tue, 11 Apr 2006 20:57:00 GMT
Server: Apache/1.3.28 (Unix) PHP/4.3.2 mod_perl/1.28
Keep-Alive: timeout=15, max=100
Content-Length: 0
ETag: "ff-2f98-3d810c51"
```

Proxy Servers
**Persistent Connection Advantages**

For HTTP/1.1, "Connection: keep-alive" is now the default. Multiple HTTP requests can be sent through the same TCP/IP connection.

**Empirical Measurements:**
- 10 documents; 1.7 kB each
- 5.3 seconds; each request a new connection
- 1.2 seconds; one connection

http://metalab.unc.edu/mdma-release/http-prob.html

---

**HTTP Cookies**

HTTP is a stateless protocol. Cookies provide a mechanism to "maintain state".

- Cookie Central: The Unofficial Cookie FAQ
  - http://www.cookiecentral.com
  - http://www.cookiecentral.com/faq

**Maintaining State with Cookies**

- HTTP State Management Mechanism
  - Cookie Central: The Unofficial Cookie FAQ
  - http://www.cookiecentral.com
  - Persistent Client State HTTP Cookies

---

**Cookie Example**

- Server returns cookie to HTTP client ("Set-Cookie" response header)
- HTTP client returns cookie to server ("Cookie" request header)

```bash
minerva% lwp-request -o http://www.espn.com/
GET http://www.espn.com/ HTTP/1.1
User-Agent: lwp-request/2.06
HTTP/1.1 301 Moved Permanently
Date: Wed, 13 Apr 2005 17:54:30 GMT
Server: Microsoft-IIS/5.0
Content-Length: 0
Cache-Control: no-cache
Pragma: no-cache
Location: http://espn.go.com/
Via: 1.1 sportrl04 (Redline Networks Web I/O Processor - T|X 3.2.18 0)
Referer: http://www.espn.com/
Accept-Language: en
Accept-Charset: ISO-8859-1, utf-8
Accept-Encoding: gzip, deflate
Accept-Encoding: gzip, deflate
Accept: text/html
Accept-Version: 1.0
Client-Date: Wed, 13 Apr 2005 17:54:31 GMT
Client-IP: 199.181.132.244:80
Client-IP: 199.181.132.244:80
P3P: CP="CAO DSP COR CURa ADMa DEVa TAIa PSAa PSDa IVAi IVDi CONi OUR SAMo OTRo BUS PHY ONL UNI PUR COM NAV INT DEM CNT STA PRE"
Set-Cookie: SWID=2DE9EC90-B800-4C6F-91C9-03EFDF70B9A0; path=/; expires=Wed, 13-Apr-2025 17:54:31 GMT; domain=.go.com;
Set-Cookie: DE2=dXNhO21hO2NhbWJyaWRnZTticm9hZGJhbmQ7NTs0OzQ7NTA2OzA0Mi4zODA7LTA3MS4xMzU7ODQwOzIyOzg4OTs2Owo=; path=/; expires=Wed, 20 Apr 2005 17:00:00 GMT; domain=.go.com
```

---

**Cookie Attributes**

- name
- expires
- domain
- path
- secure

**Cookie Example**

```bash
minerva% lwp-request -o http://www.espn.com/
GET http://www.espn.com/ HTTP/1.1
User-Agent: lwp-request/2.06
HTTP/1.1 301 Moved Permanently
Date: Wed, 13 Apr 2005 17:54:30 GMT
Server: Microsoft-IIS/5.0
Content-Length: 0
Cache-Control: no-cache
Pragma: no-cache
Location: http://espn.go.com/
Via: 1.1 sportrl04 (Redline Networks Web I/O Processor - T|X 3.2.18 0)
Referer: http://www.espn.com/
Accept-Language: en
Accept-Charset: ISO-8859-1, utf-8
Accept-Encoding: gzip, deflate
Accept-Encoding: gzip, deflate
Accept: text/html
Accept-Version: 1.0
Client-Date: Wed, 13 Apr 2005 17:54:31 GMT
Client-IP: 199.181.132.244:80
Client-IP: 199.181.132.244:80
P3P: CP="CAO DSP COR CURa ADMa DEVa TAIa PSAa PSDa IVAi IVDi CONi OUR SAMo OTRo BUS PHY ONL UNI PUR COM NAV INT DEM CNT STA PRE"
Set-Cookie: SWID=2DE9EC90-B800-4C6F-91C9-03EFDF70B9A0; path=/; expires=Wed, 13-Apr-2025 17:54:31 GMT; domain=.go.com;
Set-Cookie: DE2=dXNhO21hO2NhbWJyaWRnZTticm9hZGJhbmQ7NTs0OzQ7NTA2OzA0Mi4zODA7LTA3MS4xMzU7ODQwOzIyOzg4OTs2Owo=; path=/; expires=Wed, 20 Apr 2005 17:00:00 GMT; domain=.go.com
```

---

**Cookie Attributes**

- name
- expires
- domain
- path
- secure

**Additional Cookie Notes**

- Client: 300 total cookies
- 4 kb per cookie
- 20 cookies per server or domain

**HTTP State Management Mechanism, RFC 2965**

RFC 2109, February 1997
RFC 2965, October 2000

- name
- comment
- comment URL
- discard
- domain
- max-age
- path
- port
- secure
- version
### Cookie Example: Server Sets a Cookie

Form that will set a Cookie:
http://cscie12.dce.harvard.edu/http.cookie.cgi

Set-Cookie HTTP Response Header:

```
Set-Cookie: YourName=David%20P.%20Heitmeyer; domain=cscie12.dce.harvard.edu; path=/http/; expires=Fri, 13-May-2005 18:05:04 GMT
```

### Cookie Example: Returning a Cookie

Form that will set a Cookie:
http://cscie12.dce.harvard.edu/http.cookie.cgi

```
GET /http/cookie.cgi?name=David%20P.%20Heitmeyer HTTP/1.1
Host: cscie12.dce.harvard.edu
Connection: close
HTTP/1.1 200 OK
Connection: close
Date: Wed, 13 Apr 2005 18:11:40 GMT
Server: Apache/2.0.49 (Fedora)
Content-Type: text/html; charset=ISO-8859-1
Client-Date: Wed, 13 Apr 2005 18:11:40 GMT
Client-Peer: 140.247.197.240:80
Client-Response-Num: 1
Client-Transfer-Encoding: chunked
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html
PUBLIC "+//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US"><head><title>Form</title></head><body><h1>Hello,  David P. Heitmeyer</h1></body></html>
```

### Your Cookies

Firefox Webdeveloper Toolbar has a "Cookies" section.

### Mozilla Cookie Manager

### Cookies and Session IDs

A UserID or SessionID (a long character/number string that is uniquely assigned) is often stored in a cookie. The SessionID is used as the key or identifier when storing information about the user or session.

#### Google Cookie Example

Using Google's "Preference" page and setting:
- Search Language preference to: English, French, German
- SafeSearch Filtering: Strict Filtering
- Number of Results: 50

The Cookie name is: PREF
The Value is: 
```
ID=bb504f37cd318aa9:FF=1:LR=lang_en|lang_fr|lang_de:LD=en:NR=50:TM=1113416195:LM=1113416240:S=lurnF9ALm5Wg34rs
```
This cookie contains a session id as well as the values of certain preferences in a colon-separated data structure.
Cookies and Ad Tracking

Method: POST

Form that will set a Cookie:
http://cscie12.dce.harvard.edu/http/cookie.cgi

minerva
telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
POST /http/cookie.cgi HTTP/1.1
Host: cscie12.dce.harvard.edu
Content-Length: 10
Content-Type: application/x-www-form-urlencoded

name=David

HTTP/1.1 200 OK
Date: Wed, 13 Apr 2005 19:31:11 GMT
Server: Apache/2.0.49 (Fedora)
Set-Cookie: YourName=David; domain=cscie12.dce.harvard.edu; path=/http/; expires=Fri, 13-May-2005 19:31:20 GMT
Content-Length: 319
Connection: close
Content-Type: text/html; charset=ISO-8859-1

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en-US" xml:lang="en-US">
<head><title>Form</title></head>
<body>
<h1>Hello, David</h1>
</body></html>

WebDAV: an extension of HTTP

WebDAV stands for “Web-based Distributed Authoring and Versioning”. It is a set of extensions to the HTTP protocol which allows users to collaboratively edit and manage files on remote web servers.

HTTP Resources

- W3C HTTP
  http://www.w3.org/Protocols/
- HTTP Pocket Reference
  http://www.oreilly.com/catalog/httppe/
  by Clinton Wong (O’Reilly).
- Illustrated Guide to HTTP
  http://www.manning.com/Hethmon/
  by Paul Hethmon (Manning Publications; ISBN 0138582262)
  see sample chapters online.
- Other Readings:
  - W3C Recommendations Reduce ‘World Wide Wait’
    http://www.w3.org/Protocols/Performance.html
  - Apache Werk: HTTP version 1.1
  - WebTechniques: HTTP 1.1: What’s in it for Me?
  - Cookie Central: The Unofficial Cookie FAQ
    http://www.cookiecentral.com/faq/
Apache HTTP Server

- Apache Software Foundation
- Apache HTTP Server Project
  - Apache 1.3
  - Apache 2.0
- Apache Modules
  - PHP
  - Perl
  - Python
  - many, many others

Apache: The Most Popular Web Server on the Internet

Netcraft Web Server Survey

Tonight: Configuring Apache with .htaccess files

- Custom Error Documents
- Redirect
- Rewrite
- Directory Index
- Setting HTTP Headers
  - Expires
  - Headers
- Access Control
- Requiring a Secure Connection (SSL)

Apache Configuration Overview

- Server Configuration (.httpd.conf)
  Unless you are the server administrator, you generally will not have access to this account. On the FAS systems, you do not have read or write access to this file. Server configuration is read at server start or restart.
- Per Directory (.htaccess)
  Certain configuration directives for Apache can be placed within per-directory .htaccess files. .htaccess file is read on a per request basis.
.htaccess File Example

- ErrorDocument 404 status404.html
- Status404.html

Scope of .htaccess files

Directives within .htaccess files apply to the directory that contains the .htaccess file and all its descendants.

- Directives within the file, /home/e12/htdocs/.htaccess, would apply to all files within and "under" the public_html directory for the user cscie12.
- Directives within the file, /home/e12/htdocs/assignments/.htaccess, would apply to all files within and "under" the public_html/assignments directory for the user cscie12.

Problems You Will Have with .htaccess files

- Internal Server Error
- Can't "set" file
- Incorrect Permissions

Problems You will encounter when using .htaccess files

500 Internal Server Error

If you see begin seeing 500 Internal Server Error responses from the server after you have created or edited an .htaccess file, the most likely cause of the problem is incorrect permissions and/or an error in the directive syntax.

- Permissions on the .htaccess file are not set correctly. Just like HTML and image files, the server must be able to read the .htaccess file. The simplest way to allow that is to make your .htaccess file readable by "other".

```
minerva% pwd
/home/courses/j/h/jharvard/public_html
minerva% ls -l .htaccess
-rw-------   1 jharvard  founder         349 Nov 27 00:03 .htaccess
minerva% chmod o+r .htaccess
minerva% ls -l ~/public_html/.htaccess
-rw----r--   1 jharvard  founder         349 Nov 27 00:03 .htaccess
```

- Syntax Error. An error in the syntax of a directive in the .htaccess file will result in a 500 Internal Server Error. In addition, correct usage of a directive that is not allowed in the .htaccess file will result in a 500 status code. Whether or not a directive is allowed depends upon the server configuration file (httpd.conf; AllowOverride) and the directive itself.
Problems You will encounter when using .htaccess files

You can't "see" your .htaccess file.

- **HTTP**
  The web server is typically configured to deny requests for .htaccess files. For example, the file corresponding to the URL, http://cscie12.dce.harvard.edu/.htaccess exists and is readable by the Web server, but if we try to follow the link, we get a 403 Forbidden response.

- **UNIX**
  The `ls` command will not list files or directories that begin with a `.` (dot). In order to see the .htaccess file when you do a directory listing, use the `-a` (all) option.

- **SFTP**
  Sometimes your SFTP program will hide the "dot" files unless explicitly told to show them.

Apache Configuration Sections

Configuration directives can be limited by using "sections", such as:

- **Directory**
- **Location**
- **Files**
- **VirtualHost**
- **DirectoryMatch**
- **LocationMatch**
- **FilesMatch**

Within .htaccess

Note that only `Files` and `FilesMatch` can be used within .htaccess files.

Examples:

```html
<Files .htaccess>
  Order allow,deny
  Deny from all
</Files>
```

Examples:

```html
# deny access to any tilde backup files
<Files *
  Order Allow,deny
  Deny from all
</Files>
```

Custom Error Documents

- **Custom Error Documents**
- **Redirect**
- **Rewrite**
- **Directory Index**
- **Setting HTTP Headers**
  - `Expires` Headers
  - `Access Control`

- **Custom Error Documents**
  - `ErrorDocument 401 /apache/status401.html`
  - `ErrorDocument 403 /apache/status403.html`
  - `ErrorDocument 404 /apache/status404.html`

- **ErrorDocument directive**
  - `http://www.apache.org/docs/2.0/mod/core.html#errordocument`
  - `Custom Error Responses`
  - `http://www.apache.org/docs/2.0/custom-error.html`
HTTP Redirect

Fight Linkrot!


- Redirect
- Rewrite
- Meta http-equiv refresh

Redirecting Requests

HTTP Status Codes:
- 301 Moved permanently
- 302 Moved temporarily

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

Redirecting Requests

HTTP Status Codes:
- 301 Moved permanently
- 302 Moved temporarily

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

HTTP Redirect

Redirecting Requests

HTTP Status Codes:
- 301 Moved permanently
- 302 Moved temporarily

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource

Redirecting client requests can be very useful:
- URL moves to a new location
- resource removed
  - site structure is reorganized
- Provide "friendly" or additional URLs to access a resource
Rewrite

Rewrite uses regular expressions to match on a pattern and rewrite to a new location. For example, the Derek Bok Center site used to be a "user" account and had the "~bok_cen/" base. When moved to its own virtual host, all of the "~bok_cen/" requests could be rewritten to the new site with a single rewrite rule.

```
# rewrite for Bok Center
RewriteRule ^/~?bok_cen(.*) http://bokcenter.fas.harvard.edu$1 [R=301]
```

Examples of Rewrite Uses

Provide a standard mechanism to access course Web sites within Harvard College.

- [http://www.courses.harvard.edu/](http://www.courses.harvard.edu/)
- [http://www.courses.harvard.edu/5118](http://www.courses.harvard.edu/5118)

For example, Chemistry 7 has a catalog number of 5118, so the URL for the course Web site can be reached through:

```
http://my.harvard.edu/ich/ch/7/course=fac-chem7
```

HASCS Site Restructure

Dozens of rewrite directives were put in place when the HASCS site was restructured so that links to documents within the previous site would get redirected to the appropriate page in the new site.

Rewrite: Can be conditional

```
RewriteEngine On
RewriteCond %{HTTP_USER_AGENT} ^Lynx
RewriteRule ^(index.html)?$ text/ [R=302]
```

An aside: Text-only sites and "link"

Meta-information can be used to describe alternate content.

- [W3C Web Content Accessibility Guidelines: alternate pages](http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#alt-pages)

```
<link title="Text-only version" rel="alternate" href="http://cscie12.dce.harvard.edu/text/index.html" media="aural, braille, tty">
```

Lynx view of index.html provides the text-only version as a link:
Meta Refresh

Note: redirection may also be achieved on some browsers by using the `http-equiv` attribute of the `<meta>` element. More information and examples are provided at http://www.fas.harvard.edu/~web/tutorial/meta/refresh/. The recommended method is to do it at the server level.

<!-- in head -->
<!-- will redirect in 10 seconds -->
<meta http-equiv="Refresh" content="10; URL=http://www.harvard.edu/">

Directory Index and Listings

Note: Remember the difference between a directory having `rwx-----x` and `rwx---r-x` permissions?

- **DirectoryIndex**
  http://www.apache.org/docs/2.0/mod/mod_dir.html
- Would you prefer `main.html` or `overview.html` to be the default files returned when a directory is requested?
- **mod_autoindex**
  http://www.apache.org/docs/2.0/mod/mod_autoindex.html
  Provides for automatic indexing of a directory.

DirectoryIndex

DirectoryIndex index.html main.html overview.html slide1.html

More Control over Directory Listings

**mod_autoindex**

Basic

<table>
<thead>
<tr>
<th>Directory</th>
<th>Last Modified</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoindex</td>
<td>2009-03-31 12:30</td>
<td>203</td>
</tr>
<tr>
<td>main.html</td>
<td>2009-03-31 12:30</td>
<td>203</td>
</tr>
<tr>
<td>overview.html</td>
<td>2009-03-31 12:30</td>
<td>203</td>
</tr>
<tr>
<td>slide1.html</td>
<td>2009-03-31 12:30</td>
<td>203</td>
</tr>
</tbody>
</table>

Custom
The details:

```
minerva% pwd
/home10/c/s/cscie12/public_html/autoindex/ex2
```

```
minerva% ls -la
```

```
total 28
  drwxr-xr-x  6 cscie12  courses      8192 Nov 27 13:11 ..
  -rw-r--r--  1 cscie12  courses       207 Nov 27 13:12 .htaccess
  -rw-r--r--  1 cscie12  courses       147 Nov 27 13:09 HEADER.html
  -rw-r--r--  1 cscie12  courses       207 Nov 27 13:09 README.html
  -rw-r--r--  1 cscie12  courses       66 Nov 27 13:09 slide1.html
  -rw-r--r--  1 cscie12  courses      4168 Nov 27 12:58 slide2.html
  -rw-r--r--  1 cscie12  courses       743 Nov 27 12:58 slide3.html
```

```
minerva% cat .htaccess
IndexOptions FancyIndexing
IndexOptions IconsAreLinks NameWidth=20 ScanHTMLTitles SuppressLastModified SuppressSize SuppressColumnSorting SuppressHTMLPreamble
IndexIgnore *.gif ..
```

Setting HTTP Headers

- Expires
- Headers

Expires

```
http://www.apache.org/docs/2.0/mod/mod_expires.html
```

```
ExpiresActive On
ExpiresByType text/html   A3600
  # HTML expires in 1 hour
ExpiresByType image/gif   A2592000
  # GIF expires in 30 days
ExpiresByType image/jpeg  A2592000
  # JPEG expires in 30 days
ExpiresByType image/png   A2592000
  # PNG expires in 30 days
ExpiresDefault "now plus 1 day"
```

ExpiresActive

```
ExpiresActive On
ExpiresByType text/html   M86400
  # HTML expires 1 day after it was last modified
ExpiresByType image/gif   M86400
  # GIF expires in 1 day
ExpiresByType image/jpeg  M86400
  # JPEG expires in 1 day
ExpiresByType image/png   M86400
  # PNG expires in 1 day
ExpiresDefault "now plus 1 day"
```

Setting HTTP Headers

- Expires
- Headers

Expires

```
http://www.apache.org/docs/2.0/mod/mod_expires.html
```

```
ExpiresActive On
ExpiresByType text/html   A3600
  # HTML expires in 1 hour
ExpiresByType image/gif   A2592000
  # GIF expires in 30 days
ExpiresByType image/jpeg  A2592000
  # JPEG expires in 30 days
ExpiresByType image/png   A2592000
  # PNG expires in 30 days
ExpiresDefault "now plus 1 day"
```

ExpiresActive

```
ExpiresActive On
ExpiresByType text/html   M86400
  # HTML expires 1 day after it was last modified
ExpiresByType image/gif   M86400
  # GIF expires in 1 day
ExpiresByType image/jpeg  M86400
  # JPEG expires in 1 day
ExpiresByType image/png   M86400
  # PNG expires in 1 day
ExpiresDefault "now plus 1 day"
```

This module controls the setting of the Expires HTTP header in server responses. The expiration date can be set to be relative to either the time the source file was last modified, or to the time of the client access. The Expires HTTP header is an instruction to the client about the document's validity and persistence. If cached, the document may be fetched from the cache rather than from the source until this time has passed. After that, the cache copy is considered "expired" and invalid, and a new copy must be obtained from the source.
Do not cache

If you do not want your page cached, set these HTTP response headers:

- Cache-control: no-cache
- Pragma: no-cache
- Expires: set to now

In Apache, this would translate to:

- ExpiresDefault "now"
- Header set Pragma "no-cache"

Headers

- mod_headers
  - http://www.apache.org/docs/2.0/mod/mod_headers.html

The optional headers module allows for the customization of HTTP response headers.Headers can be merged, replaced or removed. The server will always add a "Server" and "Date" header to the HTTP response.

- Header set Author "David P. Heitmeyer"

Usertrack with Cookies

- mod_usertrack
  - http://www.apache.org/docs/2.0/mod/mod_usertrack.html

Usertrack with Cookies

```sh
using lwp-request -G http://cscie12.dce.harvard.edu/apache/usertrack/
GET http://cscie12.dce.harvard.edu/apache/usertrack/ HTTP/1.0
Connection: close
Date: Wed, 13 Apr 2005 20:32:54 GMT
Accept-Ranges: bytes
Server: Apache/2.0.49 (Fedora)
Content-Length: 59
Content-Type: text/html;charset=UTF-8
Client-Date: Wed, 13 Apr 2005 20:32:54 GMT
Client-Peer: 140.247.197.240:80
Client-Response-Num: 1

Set-Cookie2: MyCookie=140.247.197.240.1113424374035983; path=/; max-age=2858400; domain=.dce.harvard.edu; version=1
```

.htaccess file:

```sh
CookieTracking on
CookieStyle RFC2965
CookieName MyCookie
CookieExpires "1 month 3 days 2 hours"
```

WWW Access Control

You can implement access control on all or part of your web site so that:

- users must provide a username and password (Basic Authentication);
- users' computers must be within a particular domain

```sh
Usertrack with Cookies

```
Basic Authentication: Warning

Basic Authentication alone does not provide the security and privacy to adequately protect truly confidential or personal information. Basic Authentication is analogous to simply "closing a door" to parts of your Web site. It will prevent the casual or polite users from "opening the door", but will not prevent someone mildly determined to walking in.

Two issues that contribute to the lack of security and privacy are:

- the content is transmitted over the network in plaintext
- the usernames and passwords (submitted with each HTTP request) is transmitted over the network in plaintext

HTTP: Authenticate

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is '^]'.
HEAD /apache/access/example1/ HTTP/1.1
Host: cscie12.dce.harvard.edu
HTTP/1.1 401 Authorization Required
Connection: close
Date: Wed, 13 Apr 2005 20:44:39 GMT
WWW-Authenticate: Basic realm="Basic Authentication Tutorial 1"
Content-Length: 492
Content-Type: text/html; charset=iso-8859-1
Client-Date: Wed, 13 Apr 2005 20:44:39 GMT
Client-Peer: 140.247.197.240:80
Client-Response-Num: 1
```

HTTP: Authentication/Authorization

```
minerva% telnet 140.247.197.240 80
Trying 140.247.197.240...
Connected to 140.247.197.240.
Escape character is '^]'.
HEAD /apache/access/example1/ HTTP/1.1
Host: cscie12.dce.harvard.edu
Authorization: Basic Z3Vlc3Q6Z3Vlc3Q=
HTTP/1.1 200 OK
Connection: close
Date: Wed, 13 Apr 2005 20:47:53 GMT
Accept-Ranges: bytes
Server: Apache/2.0.49 (Fedora)
Content-Length: 124
Content-Type: text/html; charset=UTF-8
Client-Date: Wed, 13 Apr 2005 20:47:53 GMT
Client-Peer: 140.247.197.240:80
Client-Response-Num: 1
```

Access Control Documentation

Apache

- Apache FAQ has a section on user authentication.
- Using User Authentication from Apache Week
- Relevant Apache Module and Directive Documentation
  - mod_access module
  - mod_auth module
  - require directive
  - satisfy directive
Implementing Access Control

To implement access control, you must create a file name '.htaccess' that contains the proper configuration instructions. You may also need to create a '.htpasswd' file using the utility "htpasswd" and a '.htgroup' file.

- htpasswd program
- .htaccess file
- .htpasswd file
- .htgroup file

htpasswd file

This file contains usernames and encrypted passwords (username:enc_password). It is created and managed with the utility, "htpasswd", which can be run from the command line.

This file should not lie within your public_html. It should reside at the root level of your home directory (for example, /home/courses/harvard/.htpasswd)

This file needs to be readable by the Web Server.

Example:

```
minerva% which htpasswd
/usr/bin/htpasswd

minerva% htpasswd
Usage: htpasswd [-c] passwordfile username

The -c flag creates a new file.

minerva% htpasswd -c /var/www/public_html/.htpasswd.demo guest
Adding password for guest
New password:
*****
Re-type password:
*****
Password for "guest" (and all other entries) is "guest". Entries for guest2, guest3, and guest4 are created without the "-c" flag, since the .htpasswd.demo file already exists.

Contents of file:

guest:79WeSn3vYGsKQ
guest2:PR4APgA.4CKO.
guest3:5DbCMPbSDstj2
guest4:CjHmW1PWNFwXM
```

.htgroup file

This file contains group definitions (group_name:member1 member2 ...).

This file should not lie within your public_html. It should reside at the root level of your home directory (for example, /home/courses/harvard/.htgroup)

This file needs to be readable by the Web Server.

Example:

```
minerva% more /var/www/public_html/.htgroup.demo
VIP: guest guest4

Contents of file:

VIP: guest guest4
```

Access Control Examples

For the examples given, the user "csci12" is used. You should substitute your username and home directory appropriately.

The following .htpasswd.demo and .htgroup.demo files are used:

```
/home/courses/harvard/.htpasswd.demo
The .htpasswd.demo was generated by using the utility "htpasswd"

Contents of file:

guest:79WeSn3vYGsKQ
guest2:PR4APgA.4CKO.
guest3:5DbCMPbSDstj2
guest4:CjHmW1PWNFwXM
```
Access Control Example 1

Any valid user in .htpasswd.demo is allowed access.

The "AuthName" is the description that is displayed by the browser in the Basic Authentication dialog box.

Contents of sample .htaccess file:

```
AuthName "Basic Authentication Tutorial 1"
AuthType Basic
AuthUserFile /home/e12/.htpasswd.demo
require valid-user
```

Demonstration of Example 1

You may login as any of the following users (username:password):

- `guest:guest`
- `guest2:guest`
- `guest3:guest`
- `guest4:guest`

```bash
minerva% lwp-request -USed -C"guest:iforgot" http://cscie12.dce.harvard.edu/apache/access/example1/
GET http://cscie12.dce.harvard.edu/apache/access/example1/ --> 401 Authorization Required
```

```bash
minerva% lwp-request -USed -C"guest2:guest2" http://cscie12.dce.harvard.edu/apache/access/example1/
GET http://cscie12.dce.harvard.edu/apache/access/example1/ --> 401 Authorization Required
```

```bash
minerva% lwp-request -USed -C"guest4:guest4" http://cscie12.dce.harvard.edu/apache/access/example1/
GET http://cscie12.dce.harvard.edu/apache/access/example1/ --> 401 Authorization Required
```

Access Control Example 2

Only certain users in .htpasswd.demo are allowed access.

Contents of sample .htaccess file:

```
AuthName "Basic Authentication Tutorial 2"
AuthType Basic
AuthUserFile /home/e12/.htpasswd.demo
require user guest2 guest3
```

Demonstration of Example 2

Only guest2 and guest3 are authorized:

- `guest2:guest`
- `guest3:guest`

Unauthorized:

- `guest:guest`
- `guest4:guest`

Access Control Example 3

Only members of a particular group are allowed access.

Contents of .htaccess file:

```
AuthName "Basic Authentication Tutorial 3"
AuthType Basic
AuthUserFile /home/e12/.htpasswd.demo
AuthGroupFile /home/e12/.htgroup.demo
require group VIP
```

Contents of .htgroup.demo file:

```
VIP: guest guest4
```

Demonstration of Example 3

Only members of the group "VIP" (as defined by /home/e12/.htgroup.demo) are authorized (guest and guest4):

- `guest:guest`
- `guest4:guest`

Unauthorized:

- `guest2:guest`
- `guest3:guest`
Access Control Example 4

Only certain computers are allowed access

Contents of sample .htaccess file:

```
order deny,allow
deny from all
allow from 140.247
allow from 128.103
allow from .harvard.edu
```

Demonstration of Example 4

Computers that are on the Harvard network (computers with hostnames ending in .harvard.edu or with IP addresses beginning with 128.103 or 140.247) will have access, others will be denied.

---

Access Control Example 5

Only certain computers are denied access

Contents of sample .htaccess file:

```
order allow,deny
allow from all
deny from .fas.harvard.edu
```

Demonstration of Example 5

Connections from within the domain 'fas.harvard.edu' will be denied.

---

Access Control Example 6

Certain computers are allowed in; others must provide a username and password

Contents of sample .htaccess file:

```
order deny,allow
deny from all
allow from .yale.edu
AuthType Basic
AuthUserFile /home/e12/.htpasswd.demo
AuthName "Basic Authentication Tutorial 6"
require valid-user
satisfy any
```

Demonstration of Example 6

Connections from within "yale.edu" will be allowed; others must provide a valid username and password.

---

Access Control Example 7

Only certain computers are allowed in and users must provide a valid username and password.

Contents of sample .htaccess file:

```
order deny,allow
deny from all
allow from .harvard.edu
AuthType Basic
AuthUserFile /home/e12/.htpasswd.demo
AuthName "Basic Authentication Tutorial 7"
require valid-user
satisfy all
```

Demonstration of Example 7

Only connections from within "harvard.edu" will be allowed and users must provide a valid username and password (satisfy all).
Requiring SSL (https://)

SSL (Secure Socket Layer) is a protocol that encrypts data between the client and the server. https is HTTP over SSL. More details in our last lecture on Security and Privacy.

Contents of sample .htaccess file:

```
SSLRequireSSL
```

- Allowed: https://www.people.fas.harvard.edu/~heitmey/secure/index.html
- Forbidden: http://www.people.fas.harvard.edu/~heitmey/secure/index.html

Legal Directives I: Context

Certain Apache directives are legal within .htaccess files. Some are not.

See the Apache Documentation for details. Specifically, look at the Context line that is given for the directive in question.

- Apache Core Features
  - http://www.apache.org/docs/2.0/mod/core.html
- Apache Module List
  - http://www.apache.org/docs/2.0/mod/
- standard Apache Directives
  - http://www.apache.org/docs/2.0/mod/directives.html

The following is an excerpt from the Apache HTTP Server Version 1.3 documentation

```
ErrorDocument directive

Syntax: ErrorDocument error-code document
Context: server config, virtual host, directory, .htaccess
Status: core
Override: FileInfo
Compatibility: The directory and .htaccess contexts are only available in Apache 1.1 and later.
```

Also, the "a" indicator on the Apache Quick Reference Card indicates that the directive is valid within an .htaccess file.

Legal Directives II: AllowOverride

Users are allowed to override certain aspects of the main server configuration.

The main server configuration file (httpd.conf) contains an AllowOverride directive that determines which directives within .htaccess files Apache will process. The Override line that is given for each directive in the Apache documentation indicates which configuration directive must be active in order to use that directive with an .htaccess file.

For the FAS system, the main server configuration file has the following directive in place for users' public_html directories:

```
AllowOverride FileInfo AuthConfig Limit Indexes Options
```

The following is an excerpt from the Apache HTTP Server Version 1.3 documentation

```
ErrorDocument directive

Syntax: ErrorDocument error-code document
Context: server config, virtual host, directory, .htaccess
Status: core
Override: FileInfo
Compatibility: The directory and .htaccess contexts are only available in Apache 1.1 and later.
```

Also, the "a" indicator on the Apache Quick Reference Card indicates that the directive is valid within an .htaccess file.
### Apache Modules

On the Apache (Apache/2.0.49) minerva.dce.harvard.edu web server, the following Apache modules are active:

- `access_module` modules/mod_access.so
- `auth_module` modules/mod_auth.so
- `auth_anon_module` modules/mod_auth_anon.so
- `auth_dbm_module` modules/mod_auth_dbm.so
- `auth_digest_module` modules/mod_auth_digest.so
- `ldap_module` modules/mod_ldap.so
- `auth_ldap_module` modules/mod_auth_ldap.so
- `include_module` modules/mod_include.so
- `log_config_module` modules/mod_log_config.so
- `mod_setenvif_module` modules/mod_setenvif.so
- `mod_rewrite_module` modules/mod_rewrite.so
- `proxy_module` modules/mod_proxy.so
- `proxy_ftp_module` modules/mod_proxy_ftp.so
- `proxy_http_module` modules/mod_proxy_http.so
- `proxy_connect_module` modules/mod_proxy_connect.so
- `cache_module` modules/mod_cache.so
- `suexec_module` modules/mod_suexec.so
- `disk_cache_module` modules/mod_disk_cache.so
- `file_cache_module` modules/mod_file_cache.so
- `mem_cache_module` modules/mod_mem_cache.so
- `socgi_module` modules/mod_socgi.so

---

### Legal Directives III: Apache Modules

Apache is distributed with several modules. These modules may or may not be active within the Apache server with which you are working. The **Core features** will always be available.

For example, if the **Rewrite Module** (`mod_rewrite`) has not been activated, none of the Rewrite directives will be available to use.

Refer to the **Status** and **Module** lines in the documentation for each directive and to the documentation for the specific Apache installation you are using.