Prev Next

21.4. Apache HTTP Server Configuration

The **HTTP Configuration Tool** allows you to configure the `/etc/httpd/conf/httpd.conf` configuration file for the Apache HTTP Server. It does not use the old `srm.conf` or `access.conf` configuration files; leave them empty. Through the graphical interface, you can configure directives such as virtual hosts, logging attributes, and maximum number of connections. To start the HTTD Configuration Tool, click on **`System => Administration => Server Settings => HTTP`**.

Only modules provided with Red Hat Enterprise Linux can be configured with the **HTTP Configuration Tool**. If additional modules are installed, they can not be configured using this tool.

Caution

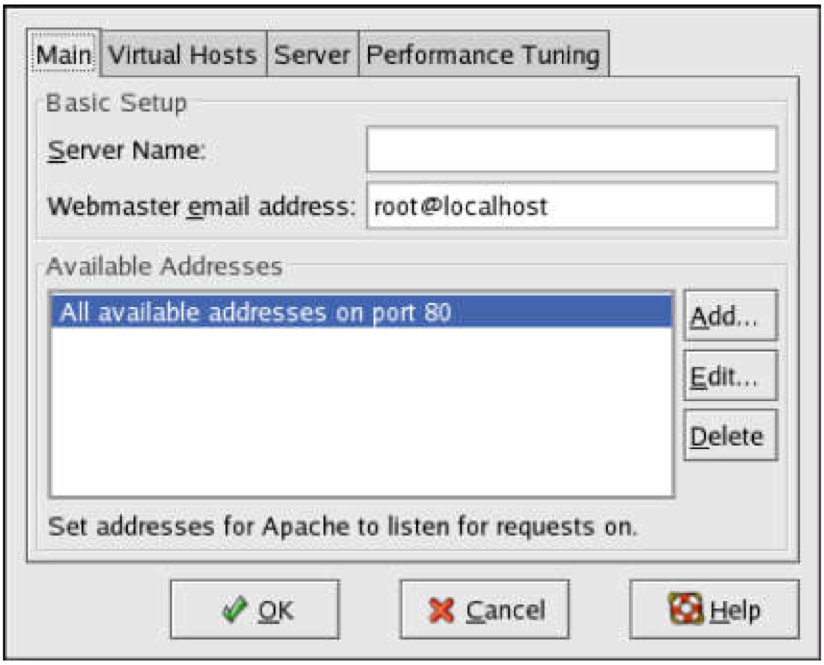
Do not edit the `/etc/httpd/conf/httpd.conf` configuration file by hand if you wish to use this tool. The **HTTP Configuration Tool** generates this file after you save your changes and exit the program. If you want to add additional modules or configuration options that are not available in **HTTP Configuration Tool**, you cannot use this tool.

The general steps for configuring the Apache HTTP Server using the **HTTP Configuration Tool** are as follows:

1. Configure the basic settings under the Main tab.
2. Click on the Virtual Hosts tab and configure the default settings.
3. Under the Virtual Hosts tab, configure the Default Virtual Host.
4. To serve more than one URL or virtual host, add any additional virtual hosts.
5. Configure the server settings under the Server tab.
6. Configure the connections settings under the Performance Tuning tab.
7. Copy all necessary files to the `DocumentRoot` and `cgi-bin` directories.
8. Exit the application and select to save your settings.

21.4.1. Basic Settings

Use the Main tab to configure the basic server settings.



Server Name:

Webmaster email address: root@localhost

All available addresses on port 80

Set addresses for Apache to listen for requests on.

Figure 21.1. Basic Settings

Enter a fully qualified domain name that you have the right to use in the Server Name text area. This option corresponds to the `ServerName` directive in `httpd.conf`. The `ServerName` directive sets the hostname of the Web server. It is used when creating redirection URLs. If you do not define a server name, the Web server attempts to resolve it from the IP address of the system. The server name does not have to be the domain name resolved from the IP address of the server. For example, you might set the server name to www.example.com while the server's real DNS name is foo.example.com.

Enter the email address of the person who maintains the Web server in the Webmaster email address text area. This option corresponds to the `ServerAdmin` directive in `httpd.conf`. If you configure the server's error pages to contain an email address, this email address is used so that users can report a problem to the server's administrator. The default value is root@localhost.

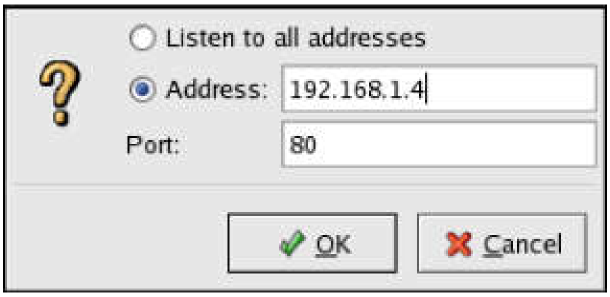
Use the Available Addresses area to define the ports on which the server accepts incoming requests. This option corresponds to the `Listen` directive in `httpd.conf`. By default, Red Hat configures the Apache HTTP Server to listen to port 80 for non-secure Web communications.

Click the Add button to define additional ports on which to accept requests. A window as shown in Figure 21.2, “Available Addresses” appears. Either choose the Listen to all addresses option to listen to all IP addresses on the defined port or specify a particular IP address over which the server accepts connections in the Address field. Only specify one IP address per port number. To specify more than one IP address with the same port number, create an entry for each IP address. If at all possible, use an IP address instead of a domain name to prevent a DNS lookup failure. Refer to http://httpd.apache.org/docs/2.2/dns-caveats.html for more information about ***Issues Regarding DNS and Apache***.

Entering an asterisk (*) in the Address field is the same as choosing Listen to all addresses . Clicking the Edit button in the Available Addresses frame shows the same window as the Add button except with the fields populated for the selected entry. To delete an entry, select it and click the Delete button.

Tip

If you set the server to listen to a port under 1024, you must be root to start it. For port 1024 and above, **`httpd`** can be started as a regular user.



○ Listen to all addresses

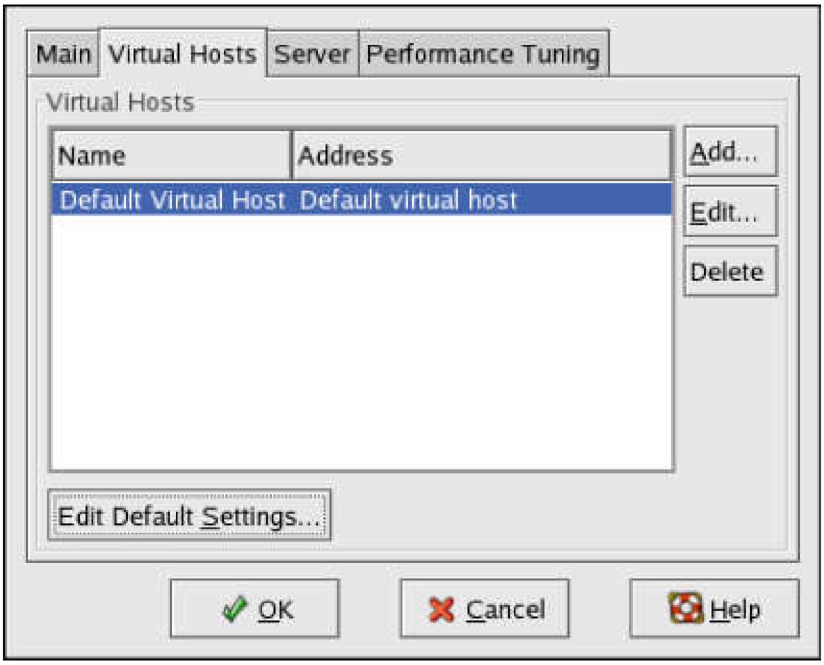
● Address: 192.168.1.4

Port: 80

Figure 21.2. Available Addresses

21.4.2. Default Settings

After defining the Server Name , Webmaster email address , and Available Addresses , click the Virtual Hosts tab. The figure below illustrates the Virtual Hosts tab.

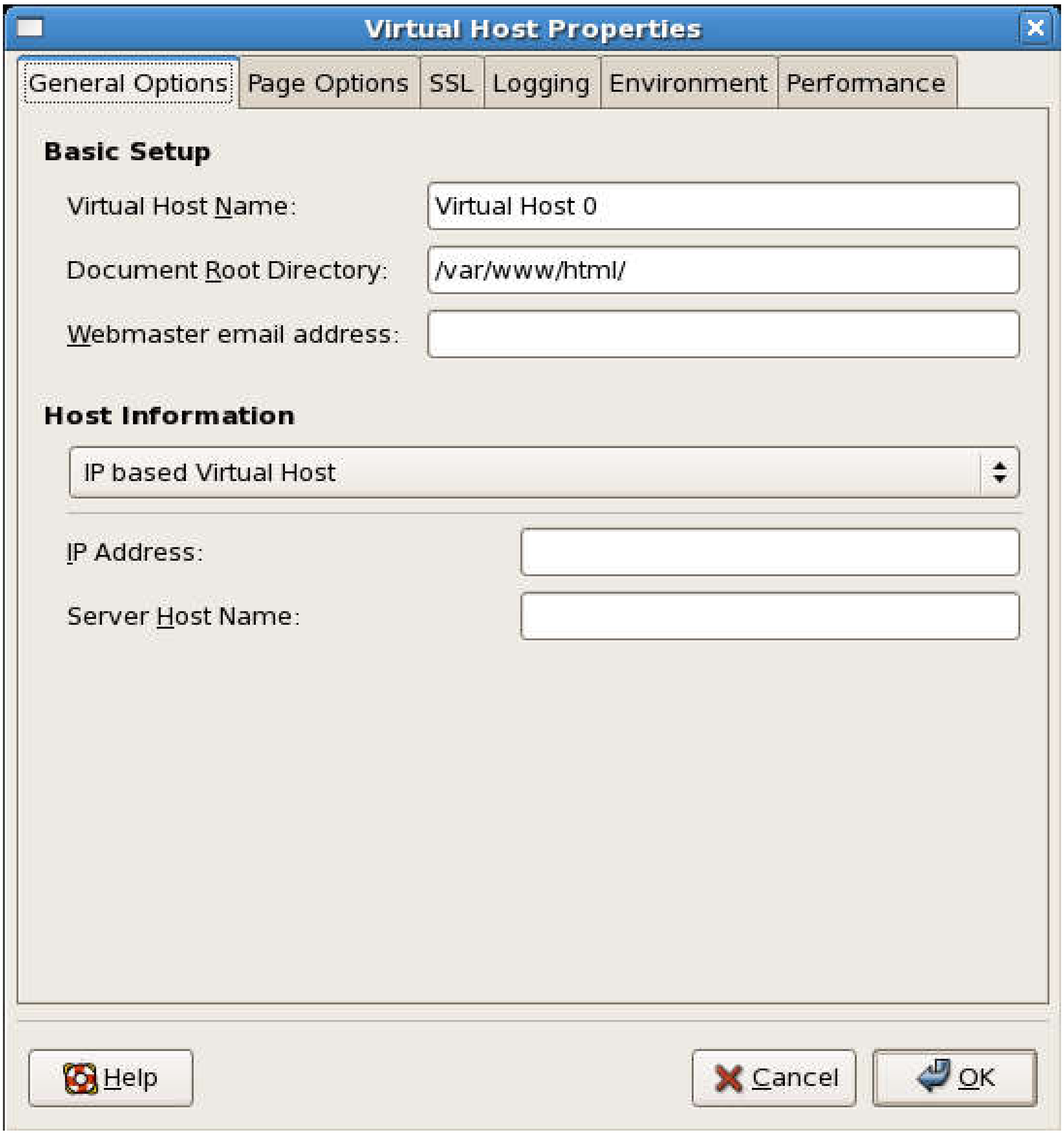


Name	Address
Default Virtual Host	Default virtual host

Figure 21.3. Virtual Hosts Tab

Clicking on Edit will display the Virtual Host Properties window from which you can set your preferred settings. To add new settings, click on the Add button which will also display the Virtual Host Properties window. Clicking on the Edit Default Settings button, displays the Virtual Host Properties window without the General Options tab.

In the General Options tab, you can change the hostname, the document root directory and also set the webmaster's email address. In the Host information, you can set the Virtual Host's IP Address and Host Name. The figure below illustrates the General Options tab.



Basic Setup

Virtual Host Name: Virtual Host 0

Document Root Directory: /var/www/html/

Webmaster email address:

Host Information

IP based Virtual Host

IP Address:

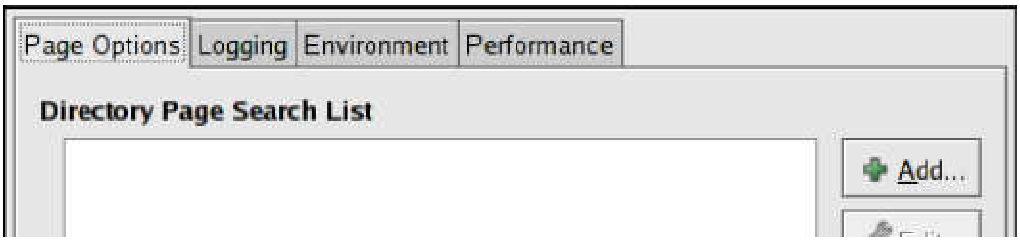
Server Host Name:

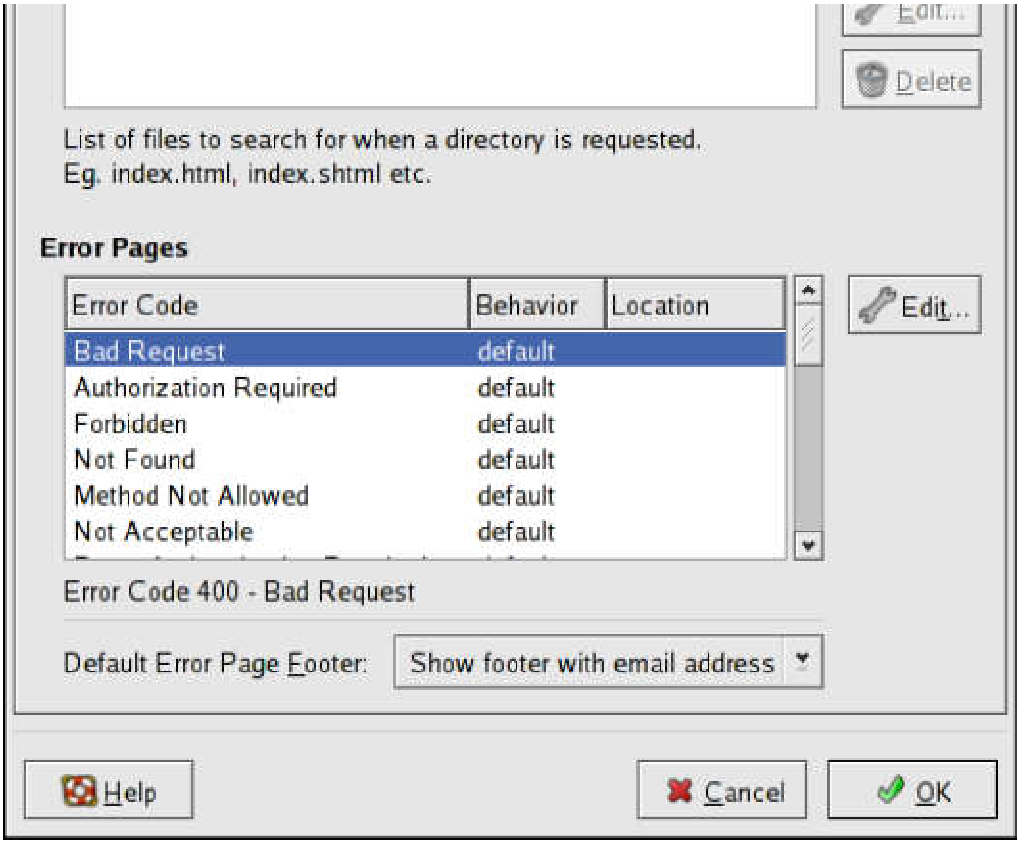
Figure 21.4. General Options

If you add a virtual host, the settings you configure for the virtual host take precedence for that virtual host. For a directive not defined within the virtual host settings, the default value is used.

21.4.2.1. Site Configuration

The figure below illustrates the Page Options tab from which you can configure the Directory Page Search List and Error Pages . If you are unsure of these settings, do not modify them.





List of files to search for when a directory is requested.
Eg. index.html, index.shtml etc.

Error Pages

Error Code	Behavior	Location
Bad Request	default	
Authorization Required	default	
Forbidden	default	
Not Found	default	
Method Not Allowed	default	
Not Acceptable	default	

Error Code 400 - Bad Request

Default Error Page Footer: Show footer with email address

Figure 21.5. Site Configuration

The entries listed in the Directory Page Search List define the `DirectoryIndex` directive. The `DirectoryIndex` is the default page served by the server when a user requests an index of a directory by specifying a forward slash (/) at the end of the directory name.

For example, when a user requests the page `http://www.example.com/this_directory/`, they are going to get either the `DirectoryIndex` page, if it exists, or a server-generated directory list. The server tries to find one of the files listed in the `DirectoryIndex` directive and returns the first one it finds. If it does not find any of these files and if `Options Indexes` is set for that directory, the server generates and returns a list, in HTML format, of the subdirectories and files in the directory.

Use the Error Code section to configure Apache HTTP Server to redirect the client to a local or external URL in the event of a problem or error. This option corresponds to the `ErrorDocument` directive. If a problem or error occurs when a client tries to connect to the Apache HTTP Server, the default action is to display the short error message shown in the Error Code column. To override this default configuration, select the error code and click the Edit button. Choose **Default** to display the default short error message. Choose **URL** to redirect the client to an external URL and enter a complete URL, including the `http://`, in the Location field. Choose **File** to redirect the client to an internal URL and enter a file location under the document root for the Web server. The location must begin the a slash (/) and be relative to the Document Root.

For example, to redirect a 404 Not Found error code to a webpage that you created in a file called `404.html`, copy `404.html` to *`DocumentRoot`*`/../error/404.html`. In this case, *`DocumentRoot`* is the Document Root directory that you have defined (the default is `/var/www/html/`). If the Document Root is left as the default location, the file should be copied to **/var/www/error/404.html**. Then, choose **File** as the Behavior for 404 - Not Found error code and enter **/error/404.html** as the **Location**.

From the Default Error Page Footer menu, you can choose one of the following options:

- Show footer with email address — Display the default footer at the bottom of all error pages along with the email address of the website maintainer specified by the **ServerAdmin** directive.
- Show footer — Display just the default footer at the bottom of error pages.
- No footer — Do not display a footer at the bottom of error pages.

21.4.2.2. SSL Support

The **mod_ssl** enables encryption of the HTTP protocol over SSL. SSL (Secure Sockets Layer) protocol is used for communication and encryption over TCP/IP networks. The SSL tab enables you to configure SSL for your server. To configure SSL you need to provide the path to your:

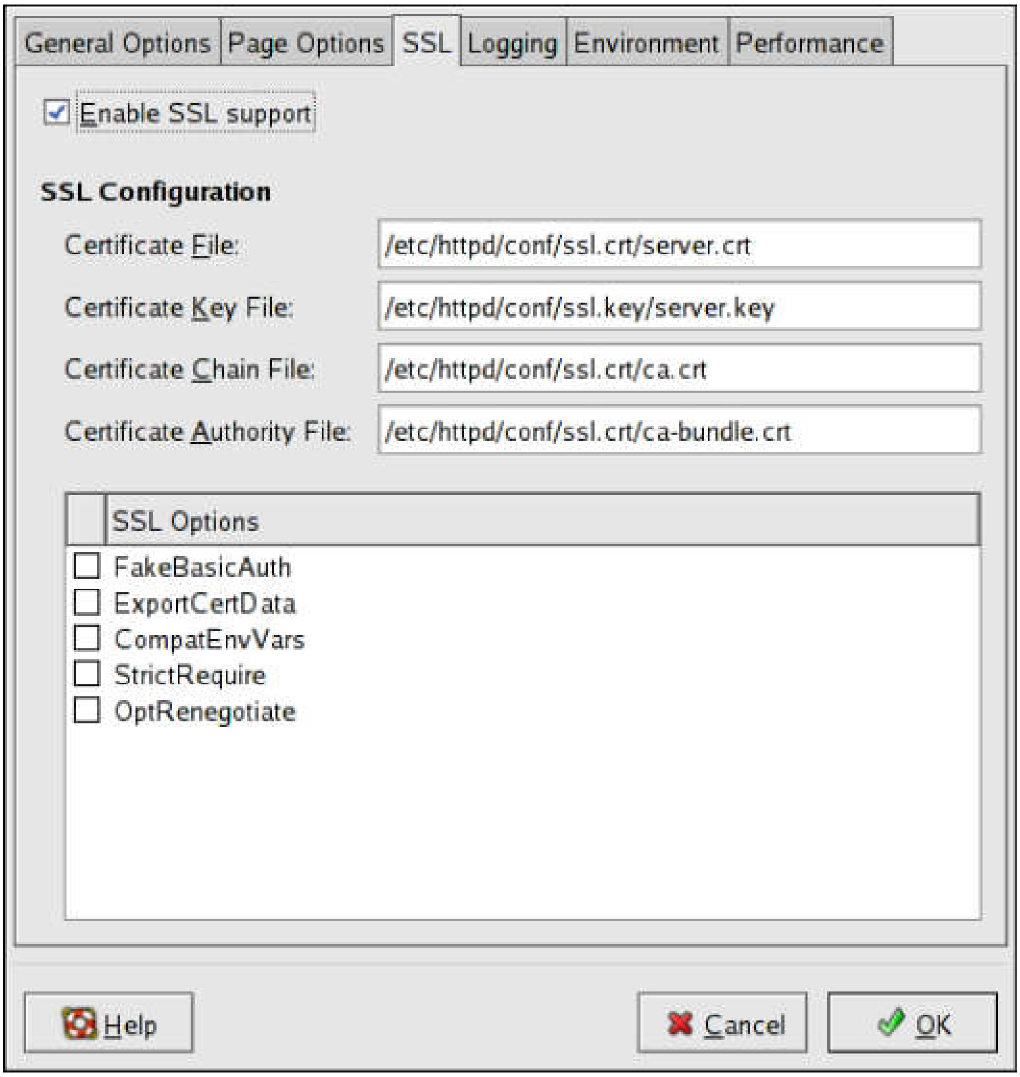
- Certificate file - equivalent to using the **SSLCertificateFile** directive which points the path to the PEM (Privacy Enhanced Mail)-encoded server certificate file.
- Key file - equivalent to using the **SSLCertificateKeyFile** directive which points the path to the PEM-encoded server private key file.
- Certificate chain file - equivalent to using the **SSLCertificateChainFile** directive which points the path to the certificate file containing all the server's chain of certificates.
- Certificate authority file - is an encrypted file used to confirm the authenticity or identity of parties communicating with the server.

You can find out more about configuration directives for SSL on http://httpd.apache.org/docs/2.2/mod/directives.html#S. You also need to determine which SSL options to enable. These are equivalent to using the **SSLOptions** with the following options:

- FakeBasicAuth - enables standard authentication methods used by Apache. This means that the Client X509 certificate's Subject Distinguished Name (DN) is translated into a basic HTTP username.
- ExportCertData - creates CGI environment variables in **SSL_SERVER_CERT**, **SSL_CLIENT_CERT** and **SSL_CLIENT_CERT_CHAIN_n** where n is a number 0,1,2,3,4... These files are used for more certificate checks by CGI scripts.
- CompatEnvVars - enables backward compatibility for Apache SSL by adding CGI environment variables.
- StrictRequire - enables strict access which forces denial of access whenever the **SSLRequireSSL** and **SSLRequire** directives indicate access is forbiden.

- OptRenegotiate - enables avoidance of unnecessary handshakes by **`mod_ssl`** which also performs safe parameter checks. It is recommended to enable OptRenegotiate on a per directory basis.

More information on the above SSL options can be found on http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#ssloptions. The figure below illustrates the SSL tab and the options discussed above.



☑ Enable SSL support

SSL Configuration

Certificate File: /etc/httpd/conf/ssl.crt/server.crt

Certificate Key File: /etc/httpd/conf/ssl.key/server.key

Certificate Chain File: /etc/httpd/conf/ssl.crt/ca.crt

Certificate Authority File: /etc/httpd/conf/ssl.crt/ca-bundle.crt

SSL Options

☐ FakeBasicAuth
☐ ExportCertData
☐ CompatEnvVars
☐ StrictRequire
☐ OptRenegotiate

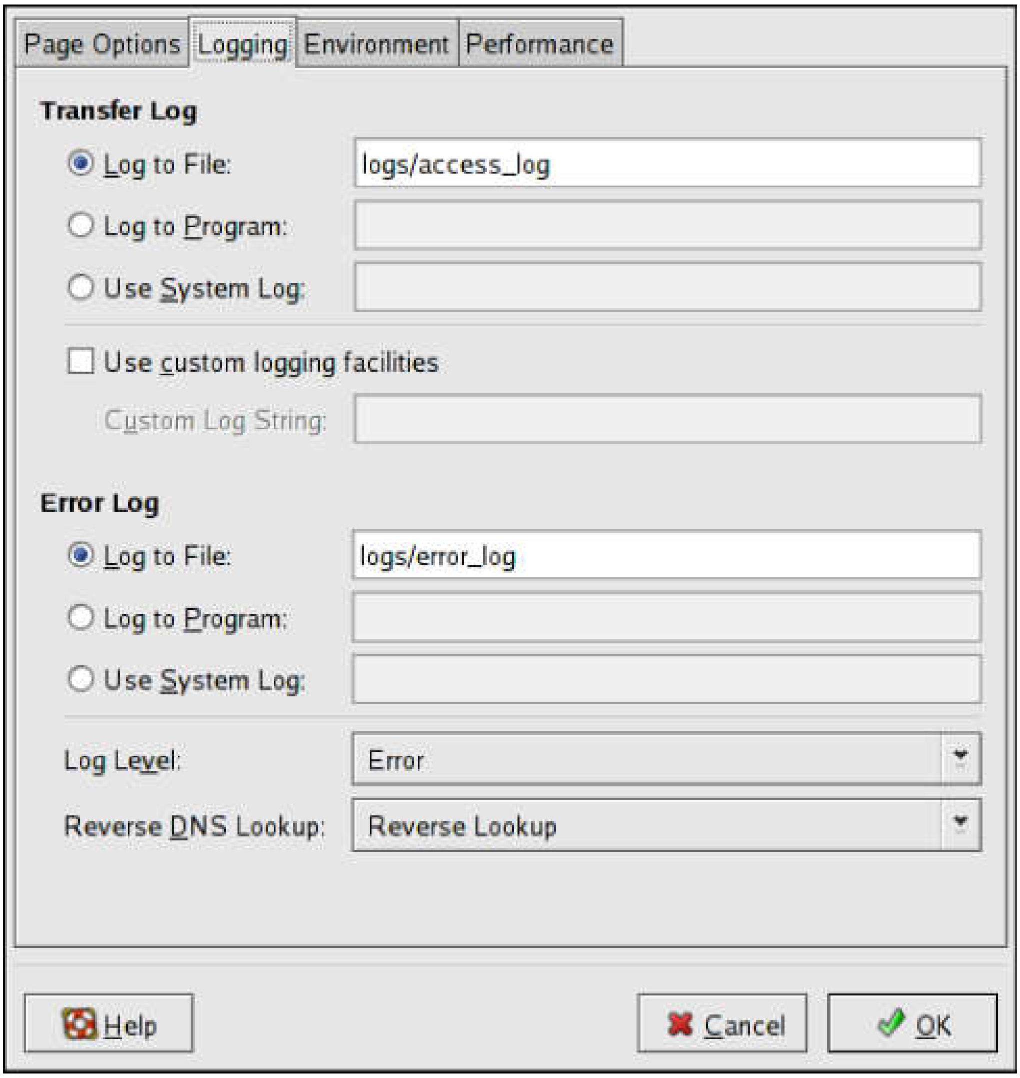
Figure 21.6. SSL

21.4.2.3. Logging

Use the Logging tab to configure options for specific transfer and error logs.

By default, the server writes the transfer log to the **`/var/log/httpd/access_log`** file and the error log to the **`/var/log/httpd/error_log`** file.

The transfer log contains a list of all attempts to access the Web server. It records the IP address of the client that is attempting to connect, the date and time of the attempt, and the file on the Web server that it is trying to retrieve. Enter the name of the path and file in which to store this information. If the path and file name do not start with a slash (/), the path is relative to the server root directory as configured. This option corresponds to the **`TransferLog`** directive.



Transfer Log

● Log to File: logs/access_log

○ Log to Program:

○ Use System Log:

□ Use custom logging facilities

Custom Log String:

Error Log

● Log to File: logs/error_log

○ Log to Program:

○ Use System Log:

Log Level: Error

Reverse DNS Lookup: Reverse Lookup

Figure 21.7. Logging

You can configure a custom log format by checking Use custom logging facilities and entering a custom log string in the Custom Log String field. This configures the **LogFormat** directive. Refer to http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#logformat for details on the format of this directive.

The error log contains a list of any server errors that occur. Enter the name of the path and file in which to store this information. If the path and file name do not start with a slash (/), the path is relative to the server root directory as configured. This option corresponds to the **ErrorLog** directive.

Use the Log Level menu to set the verbosity of the error messages in the error logs. It can be set (from least verbose to most verbose) to emerg, alert, crit, error, warn, notice, info or debug. This option corresponds to the **LogLevel** directive.

The value chosen with the Reverse DNS Lookup menu defines the **HostnameLookups** directive. Choosing No Reverse Lookup sets the value to off. Choosing Reverse Lookup sets the value to on. Choosing Double Reverse Lookup sets the value to double.

If you choose Reverse Lookup , your server automatically resolves the IP address for each connection which requests a document from your Web server. Resolving the IP address means that your server makes one or more connections to the DNS in order to find out the hostname that corresponds to a particular IP address.

If you choose Double Reverse Lookup , your server performs a double-reverse DNS. In other words, after a reverse lookup is performed, a forward lookup is performed on the result. At least one of the IP addresses in the forward lookup must match the address from the first reverse lookup.

Generally, you should leave this option set to No Reverse Lookup , because the DNS requests add a load to your server and may slow it down. If your server is busy, the effects of trying to perform these reverse lookups or double reverse lookups may be quite noticeable.

Reverse lookups and double reverse lookups are also an issue for the Internet as a whole. Each individual connection made to look up each hostname adds up. Therefore, for your own Web server's benefit, as well as for the Internet's benefit, you should leave this option set to No Reverse Lookup .

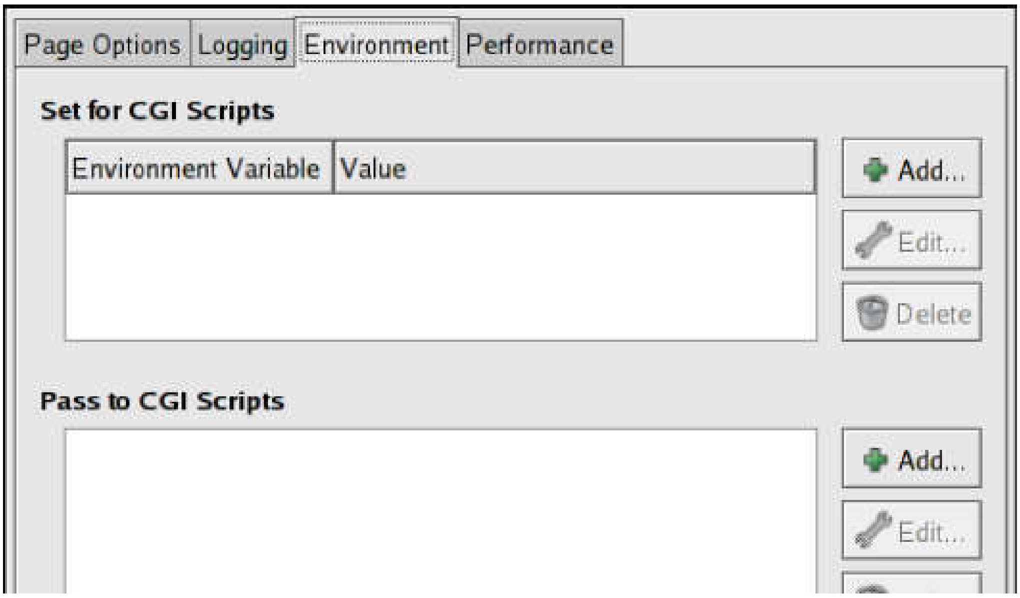
21.4.2.4. Environment Variables

Use the Environment tab to configure options for specific variables to set, pass, or unset for CGI scripts.

Sometimes it is necessary to modify environment variables for CGI scripts or server-side include (SSI) pages. The Apache HTTP Server can use the **mod_env** module to configure the environment variables which are passed to CGI scripts and SSI pages. Use the Environment Variables page to configure the directives for this module.

Use the Set for CGI Scripts section to set an environment variable that is passed to CGI scripts and SSI pages. For example, to set the environment variable **MAXNUM** to **50**, click the Add button inside the Set for CGI Script section, as shown in [Figure 21.8, "Environment Variables"], and type **MAXNUM** in the Environment Variable text field and **50** in the Value to set text field. Click OK to add it to the list. The Set for CGI Scripts section configures the **SetEnv** directive.

Use the Pass to CGI Scripts section to pass the value of an environment variable when the server is first started to CGI scripts. To see this environment variable, type the command **env** at a shell prompt. Click the Add button inside the Pass to CGI Scripts section and enter the name of the environment variable in the resulting dialog box. Click OK to add it to the list. The Pass to CGI Scripts section configures the **PassEnv** directive.



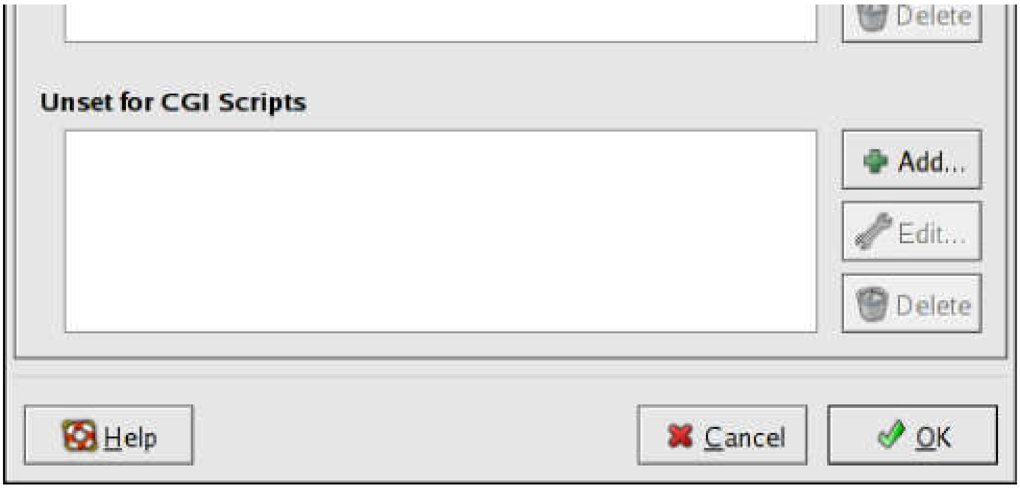


Figure 21.8. Environment Variables

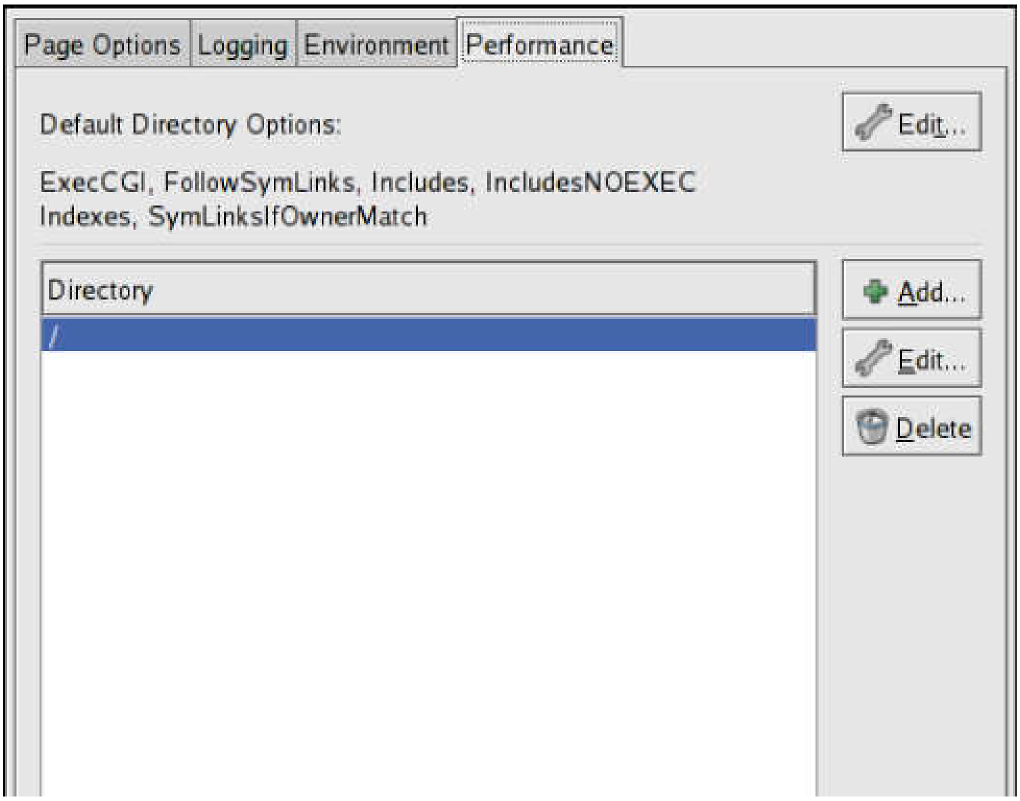
To remove an environment variable so that the value is not passed to CGI scripts and SSI pages, use the Unset for CGI Scripts section. Click Add in the Unset for CGI Scripts section, and enter the name of the environment variable to unset. Click OK to add it to the list. This corresponds to the `UnsetEnv` directive.

To edit any of these environment values, select it from the list and click the corresponding Edit button. To delete any entry from the list, select it and click the corresponding Delete button.

To learn more about environment variables in the Apache HTTP Server, refer to the following: http://httpd.apache.org/docs/2.2/env.html

21.4.2.5. Directories

Use the Directories page in the Performance tab to configure options for specific directories. This corresponds to the `<Directory>` directive.



Default Directory Options:

ExecCGI, FollowSymLinks, Includes, IncludesNOEXEC
Indexes, SymLinksIfOwnerMatch

Directory
/

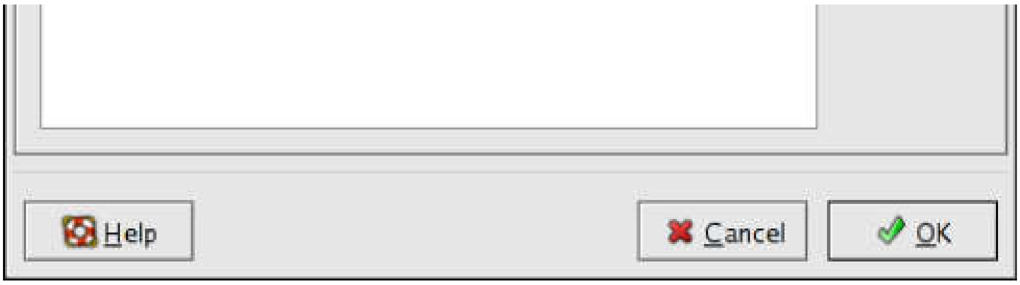


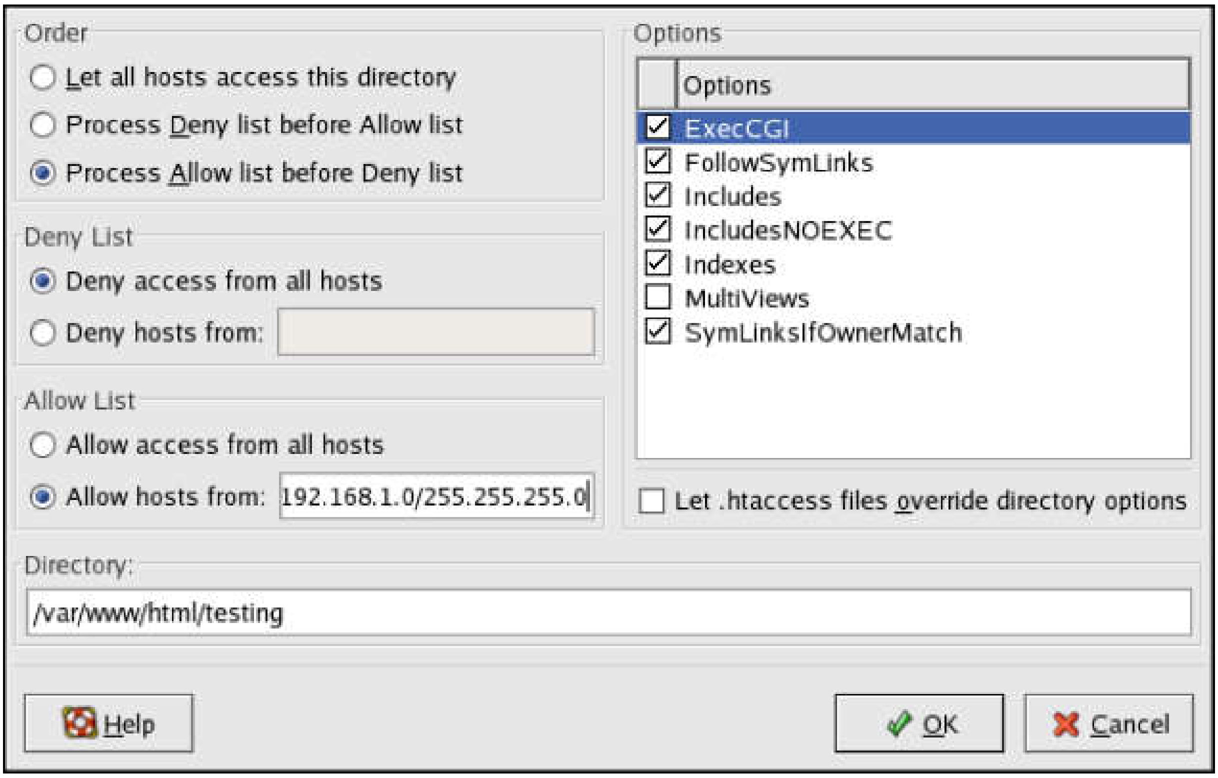
Figure 21.9. Directories

Click the Edit button in the top right-hand corner to configure the Default Directory Options for all directories that are not specified in the Directory list below it. The options that you choose are listed as the [`Options`]() directive within the [`<Directory>`]() directive. You can configure the following options:

- ExecCGI — Allow execution of CGI scripts. CGI scripts are not executed if this option is not chosen.
- FollowSymLinks — Allow symbolic links to be followed.
- Includes — Allow server-side includes.
- IncludesNOEXEC — Allow server-side includes, but disable the **`#exec`** and **`#include`** commands in CGI scripts.
- Indexes — Display a formatted list of the directory's contents, if no **`DirectoryIndex`** (such as **`index.html`**) exists in the requested directory.
- Multiview — Support content-negotiated multiviews; this option is disabled by default.
- SymLinksIfOwnerMatch — Only follow symbolic links if the target file or directory has the same owner as the link.

To specify options for specific directories, click the Add button beside the Directory list box. A window as shown in [Figure 21.10, “Directory Settings”]() appears. Enter the directory to configure in the Directory text field at the bottom of the window. Select the options in the right-hand list and configure the [`Order`]() directive with the left-hand side options. The **`Order`** directive controls the order in which allow and deny directives are evaluated. In the Allow hosts from and Deny hosts from text field, you can specify one of the following:

- Allow all hosts — Type **`all`** to allow access to all hosts.
- Partial domain name — Allow all hosts whose names match or end with the specified string.
- Full IP address — Allow access to a specific IP address.
- A subnet — Such as **`192.168.1.0/255.255.255.0`**
- A network CIDR specification — such as **`10.3.0.0/16`**



Order

○ Let all hosts access this directory

○ Process Deny list before Allow list

● Process Allow list before Deny list

Deny List

● Deny access from all hosts

○ Deny hosts from:

Allow List

○ Allow access from all hosts

● Allow hosts from: 192.168.1.0/255.255.255.0

Options

☑ ExecCGI

☑ FollowSymLinks

☑ Includes

☑ IncludesNOEXEC

☑ Indexes

□ MultiViews

☑ SymLinksIfOwnerMatch

□ Let .htaccess files override directory options

Directory: /var/www/html/testing

Figure 21.10. Directory Settings

If you check the Let .htaccess files override directory options , the configuration directives in the `.htaccess` file take precedence.

Prev
21.3. Starting and Stopping httpd
Up
Home
Next
21.5. Configuration Directives in httpd.conf