
VisNetic MailServer

Web Service Reference

Version 9.1


 powerful email server
.....
product updates: http://www.deerfield.com/products/visnetic-mailserver
other great products: http://www.deerfield.com
.....
<small>This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe criminal and civil penalties, and will be prosecuted to the maximum extent possible under law.</small>
<small>VisNetic® MailServer is a Trademark of Deerfield Communications Inc. All rights reserved. Portions Copyright© 2000-2003, IceWarp Software. VisNetic® MailServer is published by Deerfield.com®</small>

Contents

Web Service	1
Web Service	2
Web Site.....	3
Options	5
Access.....	6
Scripting.....	9
MIME	10
Documents	11
Error Responses.....	12
HTTP Header	13
Rewrite	14
Non RegEx Rewrites.....	16
Path Redirect.....	16
Host Redirect	17
Protocol Redirect.....	17
RegEx Rewrites	17
Flags	18
Server Variables.....	19
Directory Aliases.....	20
Directory Aliases - Absolute Path	21
Directory Aliases - Relative Path	21
Proxy.....	22
Proxy - General	23

Proxy - Security 24

Proxy - AntiVirus..... 25

Other 25

Re-Write Tutorial 26

 Differences 26

 Directory Aliases - Absolute Path..... 26

 Directory Aliases - Relative Path..... 27

 Non RegEx Rewrites 27

 Path Redirect 28

 Wildcard String Replace 28

 Host Redirect 29

 Protocol Redirect 29

 RegEx Rewrites..... 29

 Flags..... 30

 Server Variables 31

CHAPTER 1

Web Service

VisNetic MailServer's Web Service allows you to host websites.

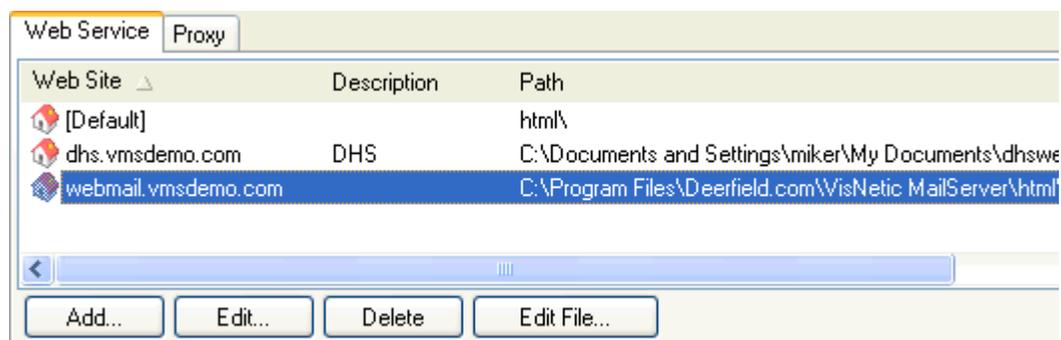
In This Chapter

Web Service	2
Proxy.....	22
Other	25
Re-Write Tutorial.....	26

CHAPTER 2

Web Service

The Web Service Tab presents you with a list of any websites you have defined to be hosted by VisNetic MailServer:



Using the **Add** or **Edit** button allows you to add or modify a site (explained in the next section).

The **Delete** button will delete the selected site definition.

The **Edit File** button will open a simple text editor allowing you to directly modify the settings for defined sites - use with care!

In This Chapter

Web Site	3
Options	5
Access	6
Scripting	9
MIME	10
Documents	11
Error Responses	12
HTTP Header	13
Rewrite	14
Directory Aliases	20

Web Site

Pressing the Add or Edit button will open the Web Site dialog

Field	Description
Active	Check this option to make this website active.
Host	The URL used to access the web service. Note that masks can be used here, for example - *.vmsdemo.com or even *.vmsdemo.* As long as the DNS records points to your server the web site will be displayed.
Description	A descriptive text for the web service.

Home directory	The root folder of the actual website files.
IP Address	If you wish you can bind this website to a specific IP address.
Use ... settings	<p>These radio buttons are only active if you are editing a new site.</p> <p>Use Default Settings</p> <p>Select this option and the new website will use the same settings as the default website.</p> <p>The default website is installed with VisNetic MailServer.</p> <p>Use Custom Settings</p> <p>Select this option if you wish to specify all options for this website yourself.</p>
Enable W3C logging	Check this box to log all connections to this website, using standardized W3C format logs.
Logging file path	<p>Specify a fully qualified file name for the log file(s).</p> <p>The variables yyyy, mm, and dd can be used within the filename:</p> <p>For example - C:\logfiles\w3clogs\"yyyymmdd\".log</p>
Delete logs older than ... days	<p>Specify the number of days after which log files will be deleted.</p> <p>A value of zero specifies that logs will never be deleted.</p>

Options

The screenshot shows the 'Web Site' Options dialog box. The 'Options' tab is active. In the 'Security' section, the following options are checked: Read, Run scripts, Directory content listing, Write, and WebDAV. The 'Execute' option is unchecked. In the 'Options' section, 'Keep open HTTP connections' is checked. The 'Maximum connections' field contains the value '127'. The 'Default virtual host' field is empty. The 'OK' and 'Cancel' buttons are located at the bottom right of the dialog.

Field	Description
Read	Check this box to allow GET and HEAD HTTP protocol requests: # GET is by far the most common method used to request a specified URL # HEAD is similar to GET but only the page headers are retrieved. This is useful for retrieving meta-information.
Scripts	Check this box to allow the execution of scripts within this website.
Directory content listing	Check this box to allow directory content listing within this website. If a folder is accessed without a page specified and VisNetic MailServer cannot find a default page (as listed under the Documents tab), then a directory listing is presented. NOTE that if you check this option, you should also specify a default virtual host on this tab.

Write	Check this box to allow # PUT HTTP protocol requests. # PUT is used to upload files to a specified Uniform Resource Identifier (URI) on the Web Server.
Executables	Check this box to allow executables (http://server.executable.cgi/exe/com) to be run on this website.
WebDAV	Check this box to allow WebDAV extensions to be used on this web site. Briefly: 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. For more information about WebDAV refer to its official portal http://www.webdav.org/ (tutorials, FAQs, ...) or see the description available within Wikipedia - http://en.wikipedia.org/wiki/WebDAV
Keep open HTTP connections	Check this options to keep a connection open for a short time after a client request. This can significantly speed up client/server communications.
Maximum connections	Specify the maximum number of simultaneous connections that you wish to allow to this website. Any requests when the limit has been reached will receive a "Server too busy" (Error 503) response.
Default virtual host	Enter the default virtual host here. This is required if you have allowed Directory content listing.

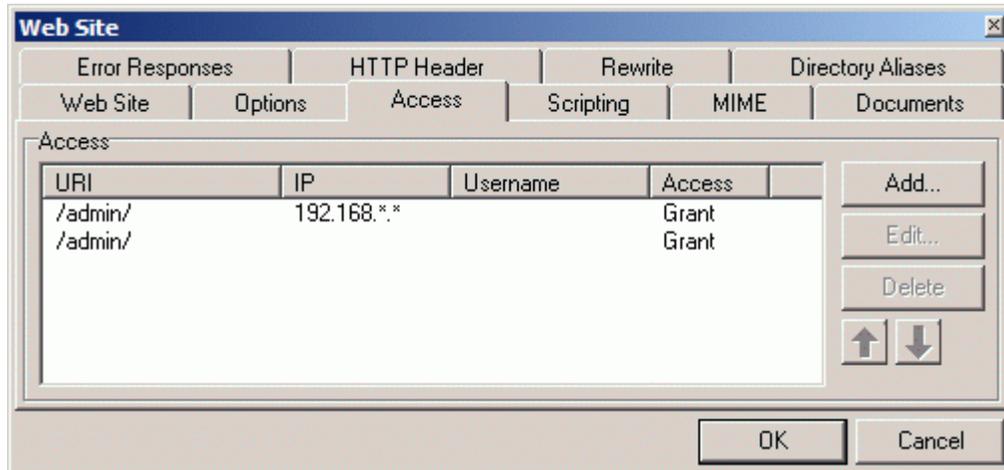
Access

The Access tab allows you to grant (allow) or deny access to a hosted website.

You can specify the whole site or individual sub-folders.

You can allow or deny access for individual IP addresses, Users (local or specifically defined), and Groups and other Account types.

Selecting the Access Tab shows you a list of currently defined access rules



Use the **Edit** and **Delete** buttons to modify or delete selected access rules.

Use the **Add** button to add an access rule, the following dialog opens:

This is where you specify the location you wish to protect and the resource(s) you are protecting it from.

You should be aware that unless you specifically Deny access to something everyone will have access. so if you want to restrict access to something to a particular user you should Deny access to everyone with one rule, then Allow access to the specific user.

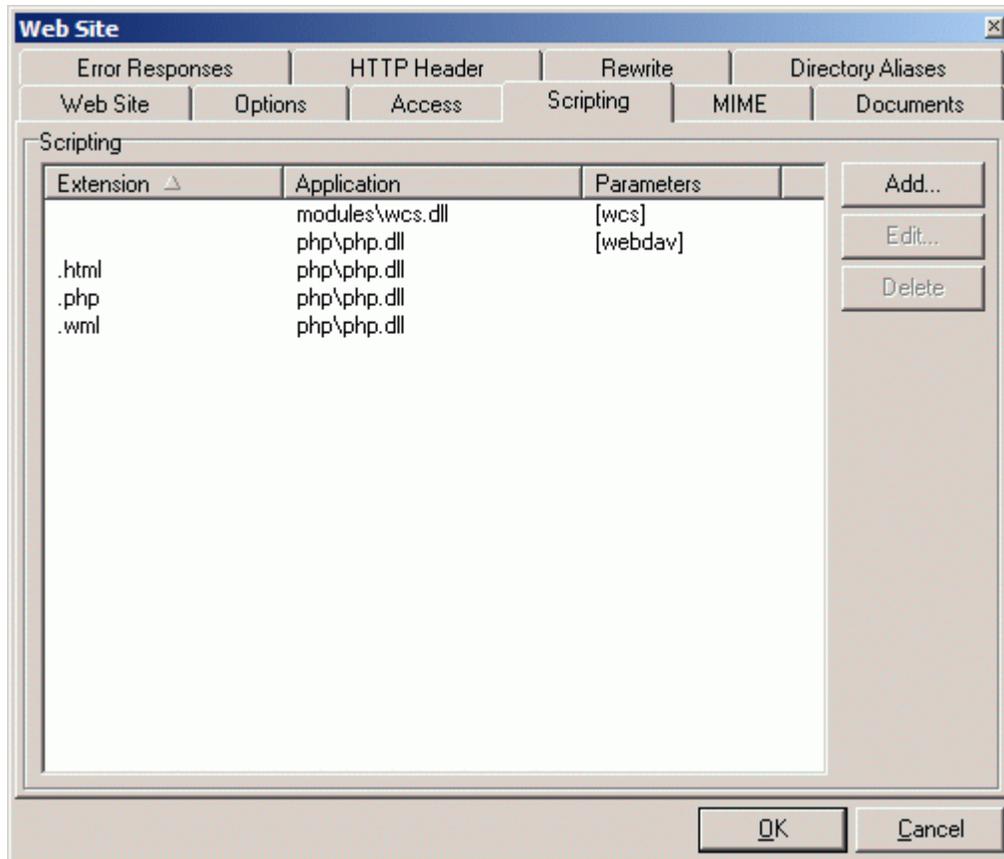
You should also be aware that if you wish to specify a local user in the Username field you should enclose it in square brackets to let VisNetic MailServer know it should check it's own database for password verification - this is done automatically if you use the "..." button to select a user, group, or domain.



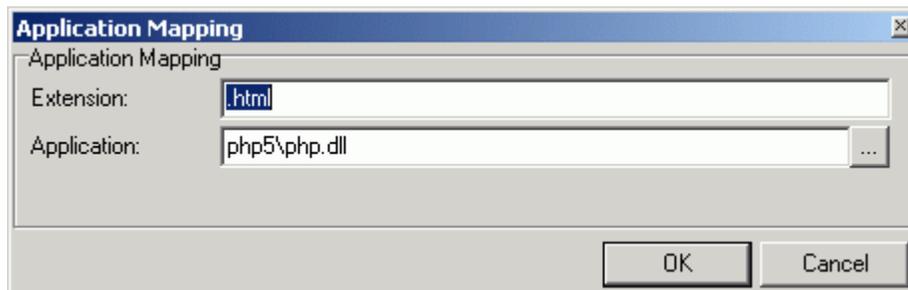
Field	Description
URI	Enter a specific URI to allow or deny access to. (optional)
IP	Enter IP address that will be allowed or denied. (optional)
Not	Check this box to logically "NOT" the Ip range. In the above example access is granted to the /admin/ directory from any IP address except 192.168.*.*
Username	Enter a specific username that will be allowed or denied. (optional)
Password	Enter a password for the username specified.
Require authenticated administrator	Check this box to allow all local System Administrator accounts access to the website with their user/password.
Require authenticated domain administrator	Check this box to allow all local System Domain Administrator accounts access to the website with their user/password.
Require authenticated system user	Check this box to allow all local System User accounts access to the website with their user/password.
Access	Choose whether access will be allowed or denied with this rule.

Scripting

Here you can specify which modules or executables should be used to process file types that the browser may not automatically understand, e.g. PHP files:



Press the Add button to link a file extension with its process application. The Application Mapping dialog opens:

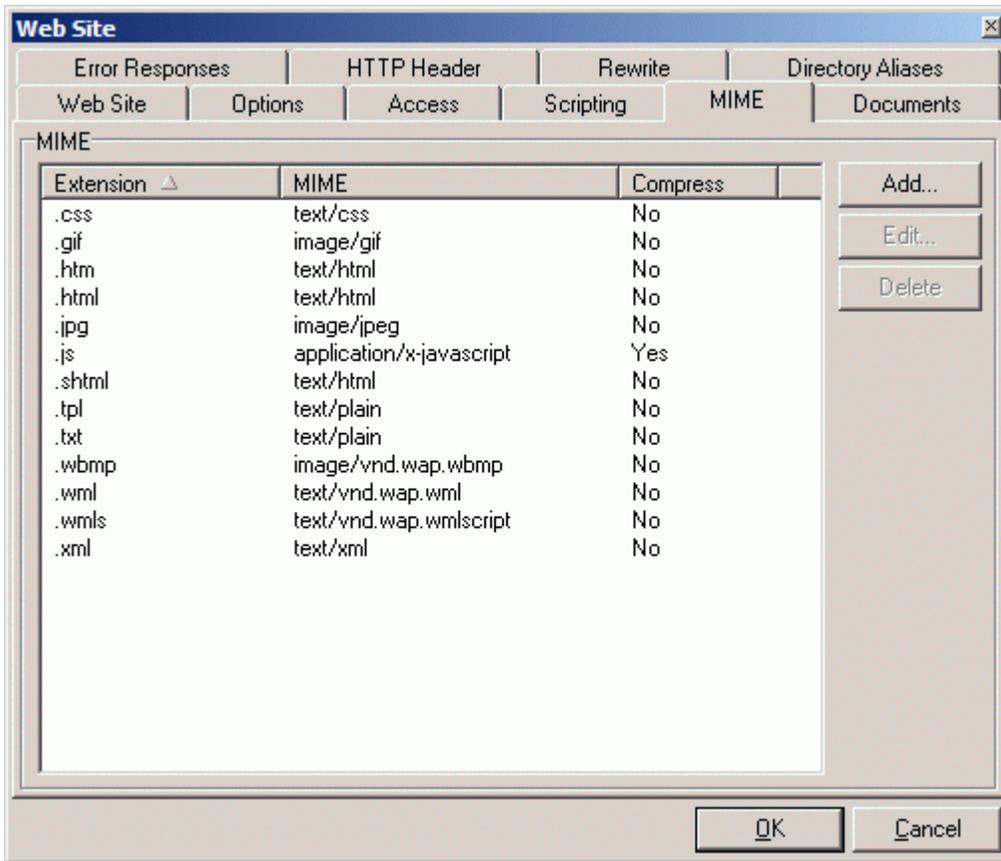


Specify the file extension (with the dot) and the full path to the Application that will process the files and press OK.

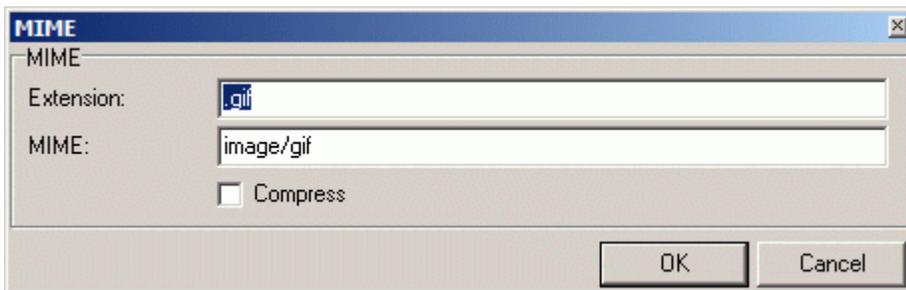
MIME

Here you can set MIME mappings for use with your web site.

There will already be a default set of mappings that should cover normal needs, but you may need to define and add your own for some purpose.



Press the Add button to add your own MIME mapping.

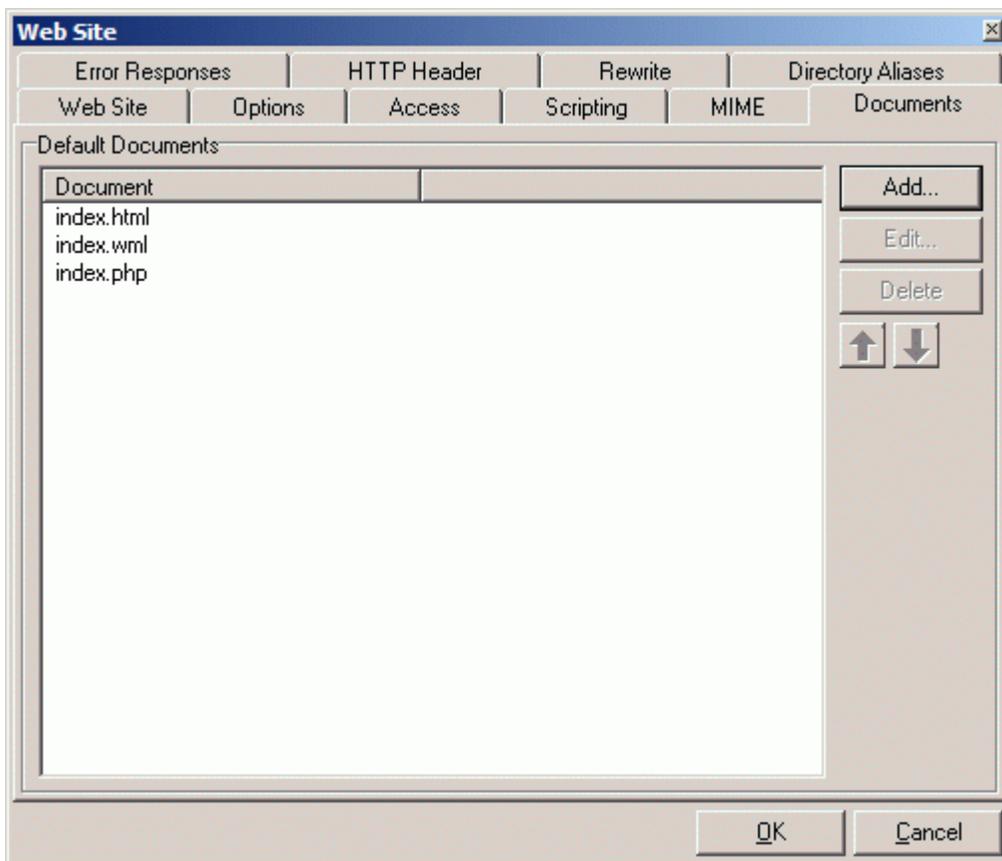


Enter the file extension you want to map, the MIME value and press OK.

You can also elect to Compress objects by checking the Compress box. If this is done then the server will GZIP the object before transferring it, provided the browser has the capability to uncompress the object.

Documents

Here you can define a list of "default" documents that the Web Server will look for if an HTTP request comes in with no specific file identified.



In the above example, if a request comes in for `http://webmail.vmsdemo.com/special` VisNetic MailServer will look for `index.html`, `index.wml` then `index.php` in the directory "special" and display the first one found.

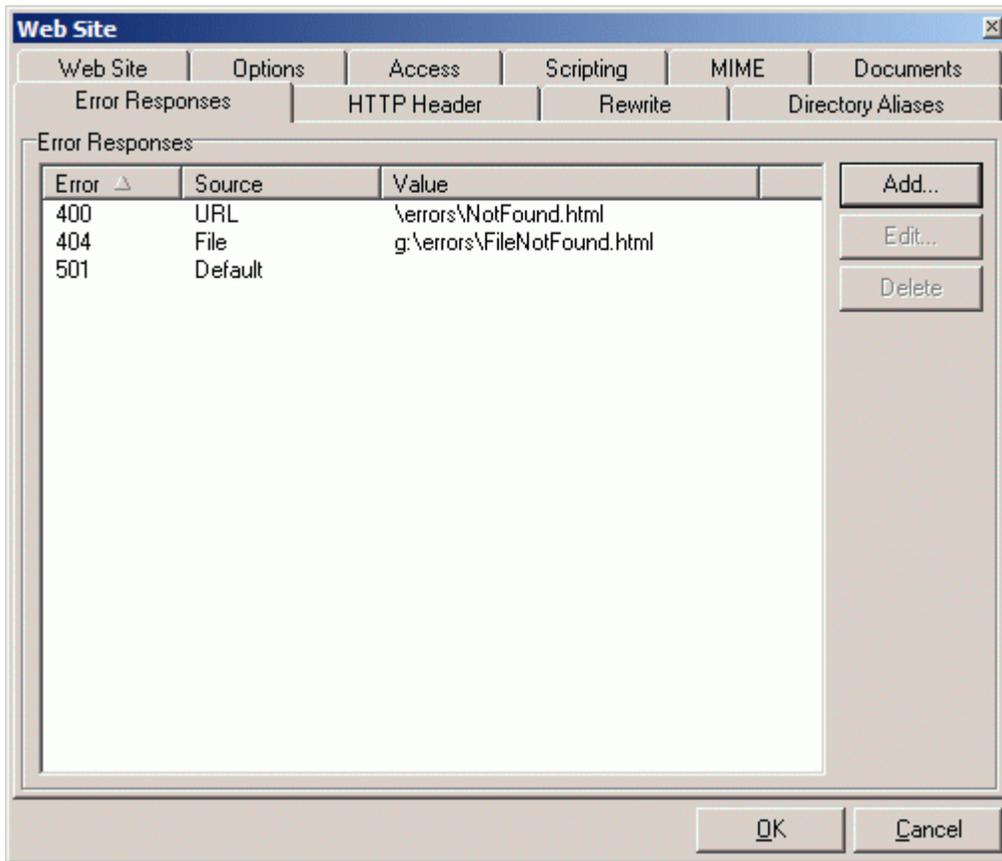
If none of the defined documents is found then VisNetic MailServer will do one of the following

If Directory content listing is allowed (see **Web Service - Options** (see "Options" on page 5)) then the directory listing for directory "special" will be displayed.

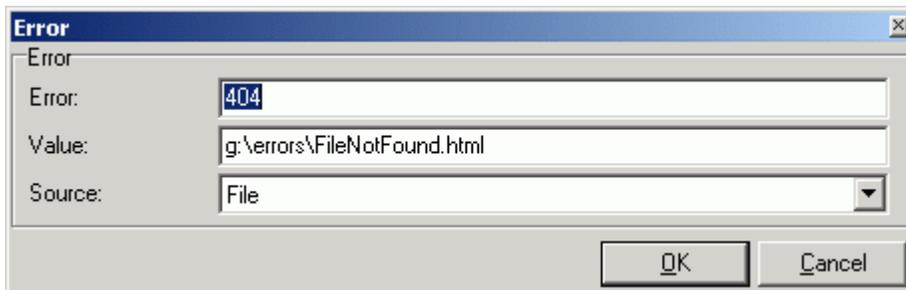
If Directory content listing is not allowed then a "Page not found" error will be returned.

Error Responses

Here you can define your own web pages to be displayed if a server Error occurs.



Press the **Add** button to define a new page for an error:



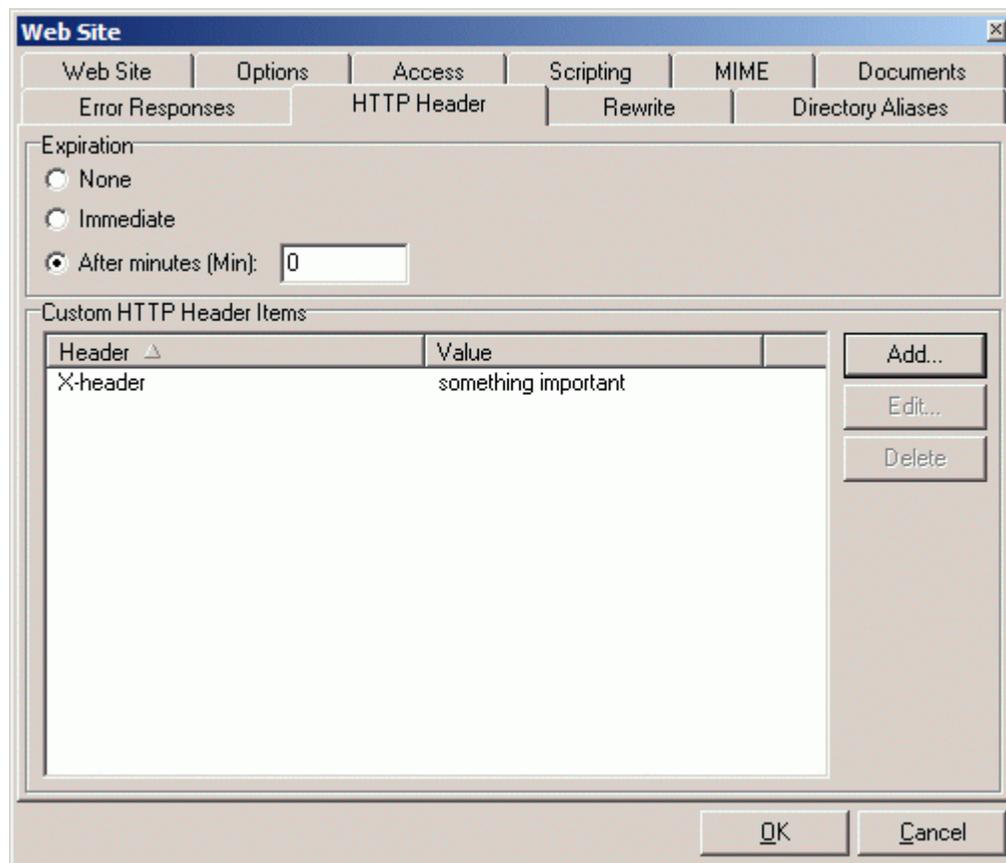
The **Error** field should contain the error code that you wish to act on.

The **Value** field should contain either fully qualified filename (if the source is a file) or a relative URL (which must be local).

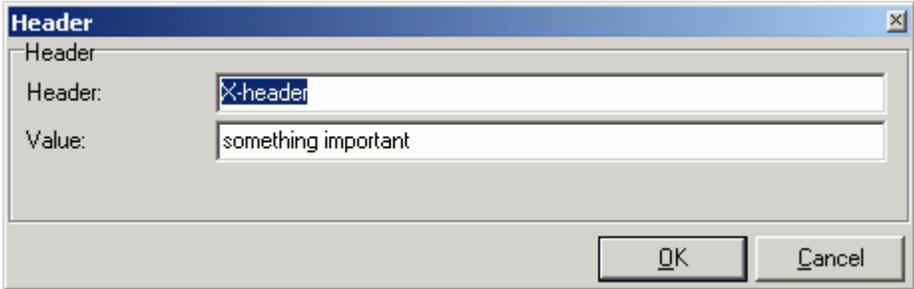
In the **Source** field you should select **File** if the Value is a filename, **URL** if the Value is a URL, or **Default** to use server default error pages.

HTTP Header

Here you can define the HTTP headers which are returned as a part of the response to a browser request.



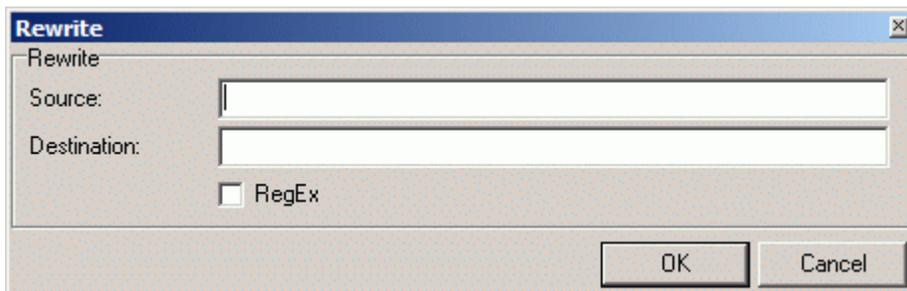
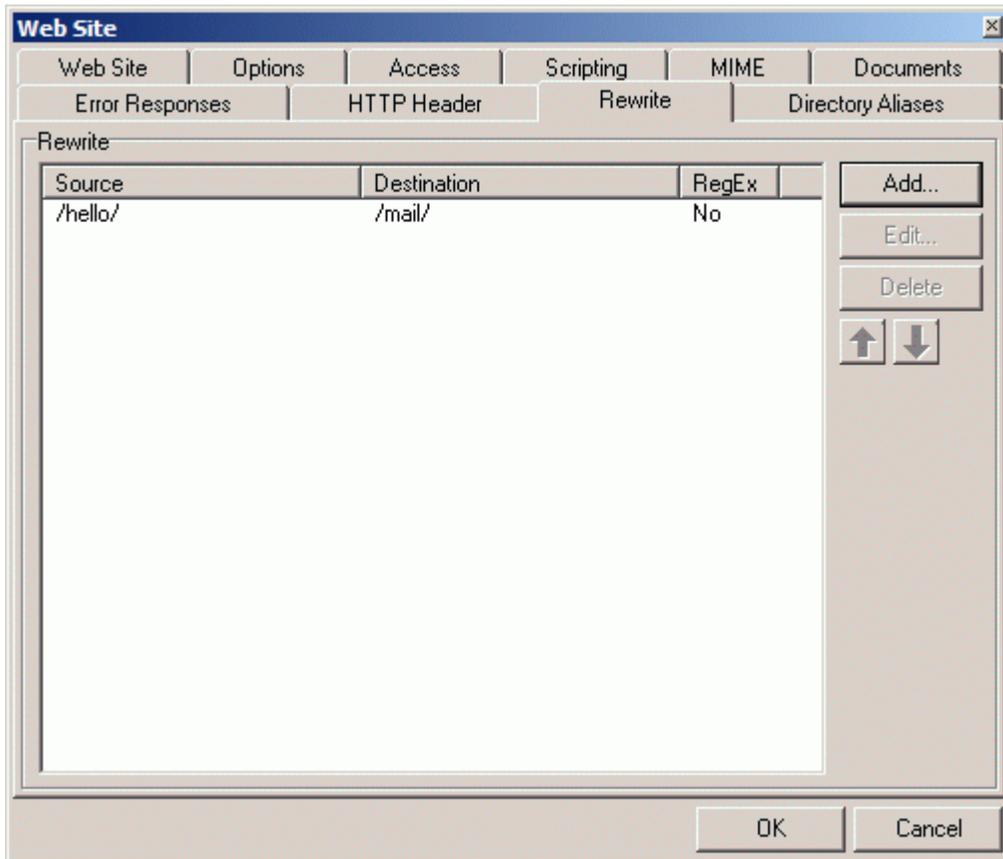
Field	Description
Expiration	<p>You can use this option to include an expiration HTTP header in the response. A browser compares the date in expiration header to the current one and decides whether the cached page should be shown or a new version should be requested.</p> <p>None The cache page would be shown if any already exists.</p> <p>Immediate</p>

	<p>The browser would have to request updated page anytime it tries to access that page.</p> <p>After minutes</p> <p>It sets the expiration period to the current time plus the number of minutes specified.</p>
Custom HTTP Header Items	<p>This allows you to define special headers.</p> <p>Press the Add button to open the Header dialog box</p>  <p>Specify the name of the header and the content you wish to insert.</p>

Rewrite

The rewrite feature is a very powerful feature allowing you to redirect requests for one URL to another URL

Assume you own mydomain.com, mydomain.net and mydomain.org. You can create one website called www.mydomain.com and redirect the .net and .org requests to the .com site.



The next section gives a tutorial on the rewrite feature.

Non RegEx Rewrites

Rewrites are more complex and flexible thus it is always hard to explain in detail. Rewrite is a feature allowing the admin to define certain rules that let him change the actual URL used or simply redirect to another URL.

There are two modes the admin can use:

Non RegEx (simple string)

RegEx (regex replace)

Let's start with the classic. Non regex is for backward compatibility and for somebody also simple as it does not require any regex knowledge. Non regex does always perform HTTP redirect. Meaning the user will see the redirect in his browser.

Support for advanced functionality has been added: non port 80, protocol redirects, wildcard string replace.

There are several types of usage here:

- Path Redirect
- Host Redirect
- Protocol Redirect

Path Redirect

`/data/ -> /otherdata/`

Eg. `http://server/data/xxx/xxx/a.txt -> http://server/otherdata/`

This would replace the data folder in the URL with /otherdata/ BUT all things coming after /data/ would not be appended to otherdata.

Wildcard String Replace

The replace does not copy the appendix data to the destination. For that you would need to use string match with * wildcard. Note that it works with relative path, such as /test/* -> /mail/*.

`vmsdemo.com* -> www.vmsdemo.com*`

`/data/* -> /otherdata/*`

Eg. `http://server/data/xxx/xxx/a.txt -> http://server/otherdata/xxx/xxx/a.txt`

Also any other combinations are possible. So you basically need to specify the asterisk in the destination too to take all remaining data from the source.

Last example we use for our web site integrated with SVN so nobody can access the .svn directories

```
*/*.* -> /
```

Eg. `http://server/mypage/.svn/...` -> `http://server/mypage/`

This makes sure that any access to a directory starting with "." will be redirected to the root of the web page. So if somebody wants to access the special `/.svn/` directory, he will get only the public content associated with the address.

Host Redirect

It is also possible to specify the name of the virtual host

```
vmsdemo.com* -> www.vmsdemo.com*
```

Eg. `http://vmsdemo.com/...` -> `http://www.vmsdemo.com/...`

The example above would simply append `www` if not specified by the user.

This would be suitable only for the primary / default virtual host, since other virtual hosts are strictly based on their hostname.

Hostname MUST BE also in the destination as in the example.

This is how it's done currently. A simple redirect like that. The same as for Path Redirect applies here, too. The difference is made by the presence of the hostname at the beginning of the Source.

Protocol Redirect

Last usage allows you to use protocol specification

```
http://www.vmsdemo.com* -> https://www.vmsdemo.com*
```

Eg. `http://vmsdemo.com/secure/*` -> `https://vmsdemo.com/secure/*`

So if a plain HTTP connection would be made to the `/secure/` URI it would be redirected to HTTPS to the same directory. It also works vice versa. From **`https://`** (`https://`) to **`http://`** (`http://`)

All combinations of these can be used for non regex Rewrite. Anything you want. I'm not aware of any other server that has this functionality.

RegEx Rewrites

The regex rewrite is in fact much simpler as there are solid rules of usage. You always work with URIs and the result is always an URI or URL for redirect.

Source is a regex match pattern and Destination is a regex replace pattern. In the destination you can also specify flags.

The whole concept is based on mod_rewrite module for Apache and uses the same syntax.

```
^/data/(.*) -> http://server/$1 [R]
```

```
http://myserver/data/other/?script=value -> http://server/other/?script=value
```

This would take the string after data, redirect to a different server, but with the selected parameters in place.

So you can see you can do some tricks with it. Every () in the regex search pattern can be then used as a variable starting with "\$" and index "n":

\$1 \$2 \$3 etc.

You can create even more sophisticated rewrites such as:

```
^/test/(.*)/(.*)$ -> /scripts/$1?value=$2
```

```
^/data/(.*)/\?(.*) -> /$1/script.asp?value=$2
```

This would not do a redirect but a simple internal URI replace. It works even with URL variables and there are no boundaries at all.

If you wish to continue with next rewrite specify the flags without [L].

```
^/data/(.*)/\?(.*) /$1/script.asp?value=$2 []
```

Also there is a special destination "-" which means not to replace anything. It might come handy sometimes.

And that's it. The rest is up to admins- look for mod_rewrite syntax for more details.

Flags

With these flags the admin can gain complete control of web server behaviour.

Flags need to be separated from the regex with space and surrounded in "[]" brackets. Such as:

[L,R]

Available flags are:

[R]edirect - redirect instead of rewrite

[L]ast - do not process other rewrite this is the last one

[F]orbidden - the user will receive 403 Forbidden message when accessed the URL

[C]hain - if the rule is not matched, skip all following rules containing [C] flag

[V=VARNAME] - match to server variable instead of URI

[] - void flag - force processing following rules

If no flag is specified, the default flag is [L]. If rewrite is matched no other rule will be processed, unless you specify void flag []. So the behavior is the same for non regex rewrites (redirects).

Server Variables

Using the [V=] flag you can achieve some sophisticated URI rewrite functionality. Instead of the URI string, the value of the server variable will be matched. Use with [C] flags and usually without the URI rewrite- thus with "-" for destination only.

Supported variables are the general HTTP_* variables: HTTP_HOST, HTTP_REFERER, HTTP_USER_AGENT, THE_REQUEST, REMOTE_IP.

```
^(www\.myhost\.com)?$ - [V=HTTP_HOST,C]
```

Virtual host is checked for "www.myhost.com".

The V= flag will be usually used with the [C] chained flag as a predecessor, such as in the following rewrite rule.

```
^/webmail/ - [C]
```

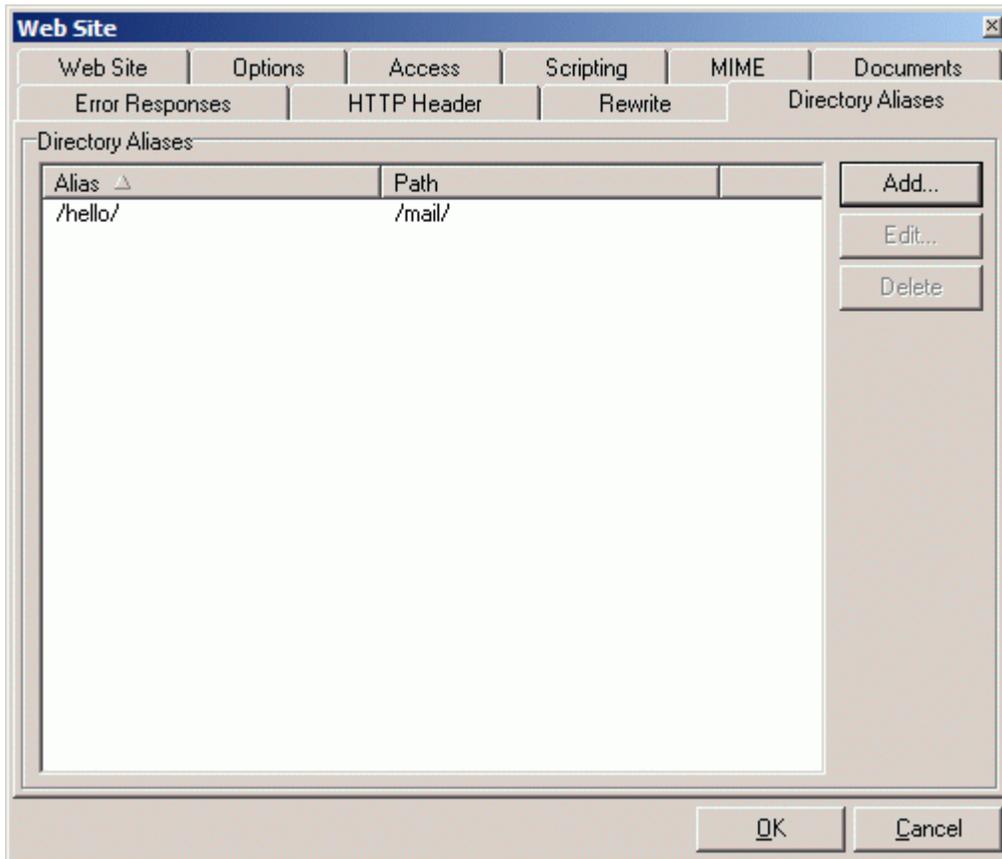
```
^Lynx/ /mail/ [V=HTTP_USER_AGENT,C,R]
```

The example above would match the /webmail/ in the URL (not replace anything) then it would check if the HTTP_USER_AGENT contains Lynx/ and if it does, redirect to /mail/. Lynx web browser simply cannot go to /webmail/ and will be redirected to /mail/

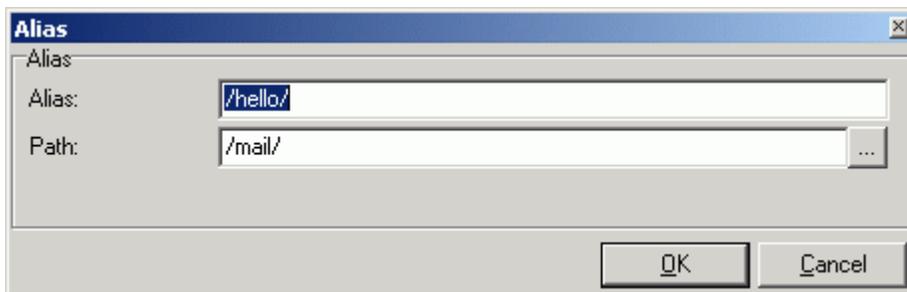
Very flexible!

Directory Aliases

The Aliases feature allows you to create aliases for directories.



Press the Add button to open the Alias dialog.



The next section gives a brief tutorial on Directory Aliases.

Directory Aliases - Absolute Path

Let me start with a short tutorial of directory aliases. There will be more examples to show what is supported and what you can do with it.

No string match functions can be used in aliases. Only the first part of the source is matched with the URI and then the path is replaced accordingly. In Absolute Path Directory Aliases, Destination is an absolute path.

```
/data/ -> /www/mydata/
```

If we have a path /data/... on the server, it will be physically loaded from /www/mydata/.... (with Linux path of course).

On Windows you could write

```
/data/ -> c:\www\data\
```

So if the the user specifies /data/ in the URL it does not have to be in the actual web sites directory but it can be loaded externally from some other location (c:\www\data\). That is what directory aliases is all about.

Directory Aliases - Relative Path

Below is an example of a relative path directory alias. In most cases you would use Rewrite feature but it is possible to use an alias for a similar functionality. In Relative Path Directory Aliases, Destination is a relative path.

```
/mail/ -> webmail/
```

This makes sure that if somebody goes to **http://server/mail/** (http://server/mail/)... it will redirect him to the webmail directory in the web site repository.

So all files loaded thru the URI /mail/... will be in fact read from /webmail/.

Notice the missing "/" at the beginning of the destination value. This is a mark of relative directory alias.

I believe it's all clear now.

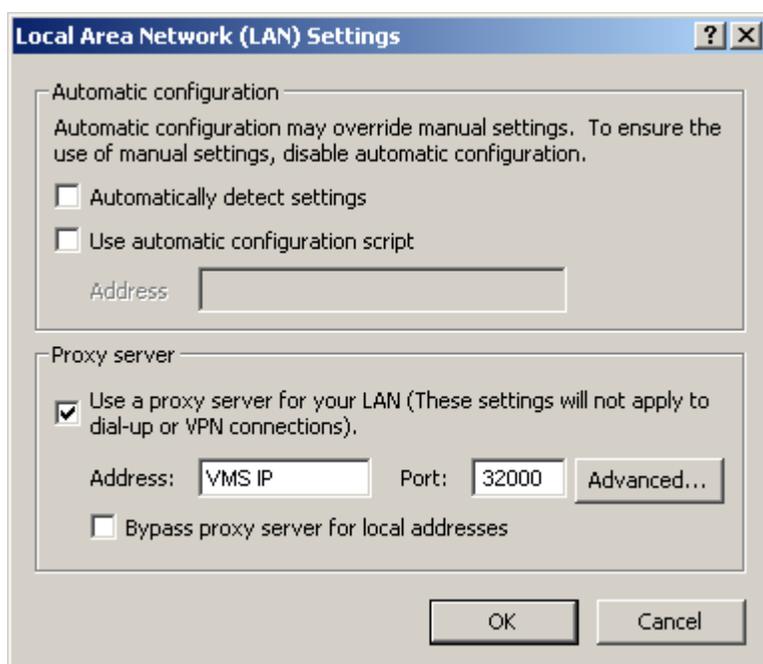
CHAPTER 3

Proxy

VisNetic MailServer has a built in Proxy server, which allows you to share Internet Browser access across your network.

Your users will need to configure their browsers to use the proxy:

Typical Browser Configuration:



VisNetic MailServer IP is the actual IP address of the server where VisNetic MailServer is running.

In This Chapter

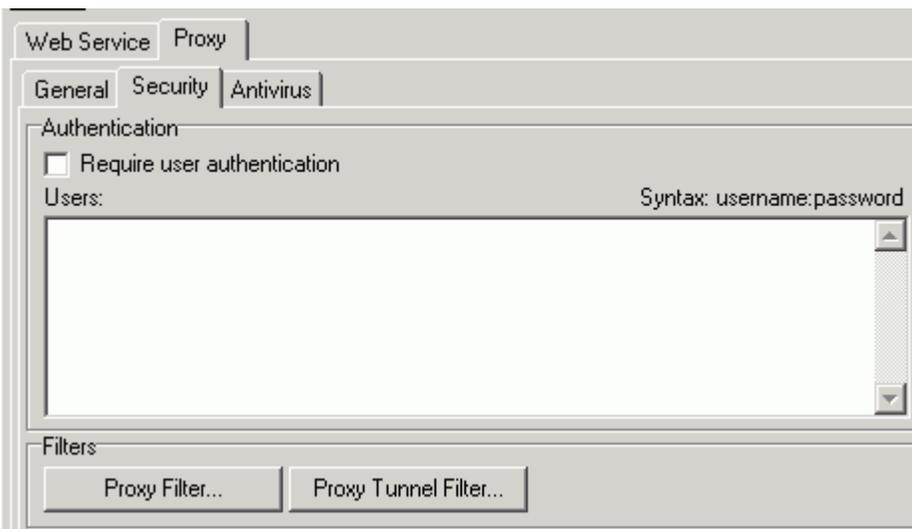
Proxy - General.....	23
Proxy - Security	24
Proxy - AntiVirus	25

Proxy - General

Field	Description
Active	Check this box to enable the proxy server.
Parent proxy	Enter the IP address of your parent proxy server here if required. This is used if your VisNetic MailServer itself connects to the internet via another proxy server.
Logging	Check this box to enable proxy logging.
Format	Select the format of the log files. Select Standard for the standard VisNetic MailServer logging format. Select W3C extended to use the extended log file format as defined by W3C
Type	Select how you want your log files: Standard creates one log file. Day creates one log file per day.
Logging path	Specify the directory to store your log files.
Delete logs older than	Specify a non-zero value to have log files deleted after the given number of days.

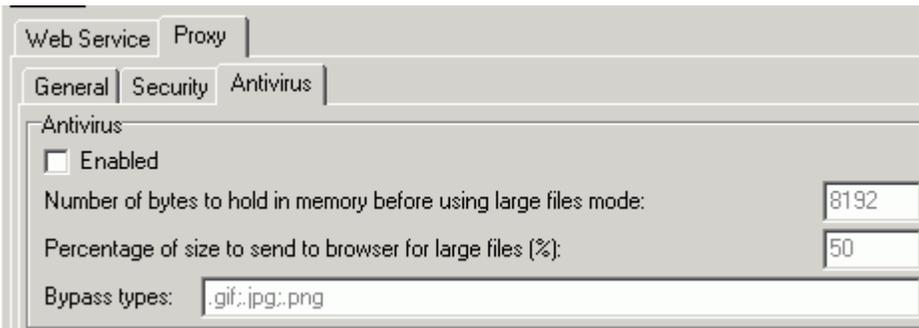
Proxy - Security

The security tab allows you to restrict access to your proxy server.



Field	Description
Require user authentication	Check this box to only allow access to the proxy server who have a specified user/password combination. User/password combinations are specified in the Users text area.
Users	Specify your list of users and passwords in the format username:password.
Proxy filter	Press the Proxy Filter button to edit a filter file where you can grant/deny access to IP ranges and hosts. Examples are given within the editor.
Proxy tunnel filter	Press the Proxy Tunnel Filter to edit filters for your local connections. Exampled are given within the editor.

Proxy - AntiVirus



Field	Description
Enabled	Enables the AntiVirus engine for the proxy server.
Number of bytes to hold in memory before using large files mode	Specifies the size data which is held in memory before sending from the server as a response. The data is held so the AntiVirus can scan the whole file. If a file is larger than this limit a special large files mode is applied. In such case the option below is used.
Percentage of size to send to browser for large files	When used the large files mode the proxy sends to browser the set number of percentage of the file and the rest is held until the complete file download. After that the proxy scans the message. If a virus is detected the remaining data is sent back filled with zeros. Otherwise the untouched remaining data is sent back.
Bypass types	You can bypass the AntiVirus engine to certain types of files. Specifies the file types with semicolon. Use only types where no viruses can be present.

Other



You can set the number of threads that are allowed to be used for the Web Service here.

CHAPTER 4

Re-Write Tutorial

In This Chapter

Differences.....	26
Directory Aliases - Absolute Path.....	26
Directory Aliases - Relative Path.....	27
Non RegEx Rewrites.....	27
RegEx Rewrites.....	29
Flags.....	30
Server Variables.....	31

Differences

The Re-write function replaces the Redirect Option found in the Web Service site section of the Console.

The Web Service - Site - Redirect option in the Administration GUI allowed you to define redirect rules based on the URL and URI. The requests that came to the server based on some string criteria were redirected to other pages. This option has been renamed to Rewrite and currently supports the old functionality for backward compatibility and a new mod_rewrite regex replace mechanism.

Non-RegEx (old way) does always redirect and the user can see the change in browser address bar. RegEx way rewrites the URL internally so for the user the URL appears to be the one shown in address bar.

Alias option has been renamed to Directory Aliases and is strictly for virtual directories. This functionality remains the same as used to be but is a bit simplified, updated and working in the end. You can have a relative alias (pointing to the current web site repository, /mail/ -> webmail/) or absolute pointing to any directory or disk on your computer - /data/ -> /www/mydata/. Subdirectories are also supported, there's a new no match strings function.

Changes summarized:

- Rewrite without RegEx does the former redirect - thus Redirect can be achieved.
- Rewrite with RegEx does rewrite and supports mod_rewrite options.
- Aliases are only directory aliases and nothing else. Support absolute and relative paths.

Directory Aliases - Absolute Path

Let me start with a short tutorial of directory aliases. There will be more examples to show what is supported and what you can do with it.

No string match functions can be used in aliases. Only the first part of the source is matched with the URI and then the path is replaced accordingly. In Absolute Path Directory Aliases, Destination is an absolute path.

```
/data/ -> /www/mydata/
```

If we have a path /data/... on the server, it will be physically loaded from /www/mydata/.... (with Linux path of course).

On Windows you could write

```
/data/ -> c:\www\data\
```

So if the the user specifies /data/ in the URL it does not have to be in the actual web sites directory but it can be loaded externally from some other location (c:\www\data\). That is what directory aliases is all about.

Directory Aliases - Relative Path

Below is an example of a relative path directory alias. In most cases you would use Rewrite feature but it is possible to use an alias for a similar functionality. In Relative Path Directory Aliases, Destination is a relative path.

```
/mail/ -> webmail/
```

This makes sure that if somebody goes to **http://server/mail/** (http://server/mail/)... it will redirect him to the webmail directory in the web site repository.

So all files loaded thru the URI /mail/... will be in fact read from /webmail/.

Notice the missing "/" at the beginning of the destination value. This is a mark of relative directory alias.

I believe it's all clear now.

Non RegEx Rewrites

Rewrites are more complex and flexible thus it is always hard to explain in detail. Rewrite is a feature allowing the admin to define certain rules that let him change the actual URL used or simply redirect to another URL.

There are two modes the admin can use:

Non RegEx (simple string)

RegEx (regex replace)

Let's start with the classic. Non regex is for backward compatibility and for somebody also simple as it does not require any regex knowledge. Non regex does always perform HTTP redirect. Meaning the user will see the redirect in his browser.

Support for advanced functionality has been added: non port 80, protocol redirects, wildcard string replace.

There are several types of usage here:

- Path Redirect
- Host Redirect
- Protocol Redirect

Path Redirect

`/data/ -> /otherdata/`

Eg. `http://server/data/xxx/xxx/a.txt -> http://server/otherdata/`

This would replace the data folder in the URL with /otherdata/ BUT all things coming after /data/ would not be appended to otherdata.

Wildcard String Replace

The replace does not copy the appendix data to the destination. For that you would need to use string match with * wildcard. Note that it works with relative path, such as /test/* -> /mail/*.

`vmsdemo.com* -> www.vmsdemo.com*`

`/data/* -> /otherdata/*`

Eg. `http://server/data/xxx/xxx/a.txt -> http://server/otherdata/xxx/xxx/a.txt`

Also any other combinations are possible. So you basically need to specify the asterisk in the destination too to take all remaining data from the source.

Last example we use for our web site integrated with SVN so nobody can access the .svn directories

`*/*. * -> /`

Eg. `http://server/mypage/.svn/... -> http://server/mypage/`

This makes sure that any access to a directory starting with "." will be redirected to the root of the web page. So if somebody wants to access the special /svn/ directory, he will get only the public content associated with the address.

Host Redirect

It is also possible to specify the name of the virtual host

```
vmsdemo.com* -> www.vmsdemo.com*
```

Eg. `http://vmsdemo.com/...` -> `http://www.vmsdemo.com/...`

The example above would simply append `www` if not specified by the user.

This would be suitable only for the primary / default virtual host, since other virtual hosts are strictly based on their hostname.

Hostname MUST BE also in the destination as in the example.

This is how it's done currently. A simple redirect like that. The same as for Path Redirect applies here, too. The difference is made by the presence of the hostname at the beginning of the Source.

Protocol Redirect

Last usage allows you to use protocol specification

```
http://www.vmsdemo.com* -> https://www.vmsdemo.com*
```

Eg. `http://vmsdemo.com/secure/*` -> `https://vmsdemo.com/secure/*`

So if a plain HTTP connection would be made to the `/secure/` URI it would be redirected to HTTPS to the same directory. It also works vice versa. From ***https://*** (`https://`) to ***http://*** (`http://`)

All combinations of these can be used for non regex Rewrite. Anything you want. I'm not aware of any other server that has this functionality.

RegEx Rewrites

The regex rewrite is in fact much simpler as there are solid rules of usage. You always work with URIs and the result is always an URI or URL for redirect.

Source is a regex match pattern and Destination is a regex replace pattern. In the destination you can also specify flags.

The whole concept is based on `mod_rewrite` module for Apache and uses the same syntax.

```
^/data/(.*) -> http://server/$1 [R]
```

```
http://myserver/data/other/?script=value -> http://server/other/?script=value
```

This would take the string after data, redirect to a different server, but with the selected parameters in place.

So you can see you can do some tricks with it. Every () in the regex search pattern can be then used as a variable starting with "\$" and index "n":

```
$1 $2 $3 etc.
```

You can create even more sophisticated rewrites such as:

```
^/test/(.*)/(.*)$ -> /scripts/$1?value=$2
```

```
^/data/(.*)/\?(.*) -> /$1/script.asp?value=$2
```

This would not do a redirect but a simple internal URI replace. It works even with URL variables and there are no boundaries at all.

If you wish to continue with next rewrite specify the flags without [L].

```
^/data/(.*)/\?(.*) /$1/script.asp?value=$2 []
```

Also there is a special destination "-" which means not to replace anything. It might come handy sometimes.

And that's it. The rest is up to admins- look for mod_rewrite syntax for more details.

Flags

With these flags the admin can gain complete control of web server behaviour.

Flags need to be separated from the regex with space and surrounded in "[]" brackets. Such as:

```
[L,R]
```

Available flags are:

[R]edirect - redirect instead of rewrite

[L]ast - do not process other rewrite this is the last one

[F]orbidden - the user will receive 403 Forbidden message when accessed the URL

[C]hain - if the rule is not matched, skip all following rules containing [C] flag

[V=VARNAME] - match to server variable instead of URI

[] - void flag - force processing following rules

If no flag is specified, the default flag is [L]. If rewrite is matched no other rule will be processed, unless you specify void flag []. So the behavior is the same for non regex rewrites (redirects).

Server Variables

Using the [V=] flag you can achieve some sophisticated URI rewrite functionality. Instead of the URI string, the value of the server variable will be matched. Use with [C] flags and usually without the URI rewrite- thus with "-" for destination only.

Supported variables are the general HTTP_* variables: HTTP_HOST, HTTP_REFERER, HTTP_USER_AGENT, THE_REQUEST, REMOTE_IP.

```
^(www\.myhost\.com)?$ - [V=HTTP_HOST,C]
```

Virtual host is checked for "www.myhost.com".

The V= flag will be usually used with the [C] chained flag as a predecessor, such as in the following rewrite rule.

```
^/webmail/ - [C]
```

```
^Lynx/ /mail/ [V=HTTP_USER_AGENT,C,R]
```

The example above would match the /webmail/ in the URL (not replace anything) then it would check if the HTTP_USER_AGENT contains Lynx/ and if it does, redirect to /mail/. Lynx web browser simply cannot go to /webmail/ and will be redirected to /mail/

Very flexible!

Index

A

Access • 6

D

Differences • 26

Directory Aliases • 20

Directory Aliases - Absolute Path • 21, 26

Directory Aliases - Relative Path • 21, 27

Documents • 11

E

Error Responses • 12

F

Flags • 18, 30

H

Host Redirect • 17, 29

HTTP Header • 13

M

MIME • 10

N

Non RegEx Rewrites • 16, 27

O

Options • 5, 11

Other • 25

P

Path Redirect • 16, 28

Protocol Redirect • 17, 29

Proxy • 22

Proxy - AntiVirus • 25

Proxy - General • 23

Proxy - Security • 24

R

RegEx Rewrites • 17, 29

Rewrite • 14

Re-Write Tutorial • 26

S

Scripting • 9

Server Variables • 19, 31

W

Web Service • 1, 2

Web Site • 3

Wildcard String Replace • 16, 28