

## **Setup of Cobalt RaQ Server for Use with Actinic Host**

**© Actinic Software Ltd.**

Author: Shaun Sloan  
Date: 15<sup>th</sup> August 2001  
Revision: 1.4

## Revision History

Revision	Date	Author	Comments
1.0	6 <sup>th</sup> April 2001	Shaun Sloan	Initial revision
1.1	9 <sup>th</sup> April 2001	Shaun Sloan	Added "regex" explanation
1.2	20 <sup>th</sup> April 2001	Shaun Sloan	Updated directory information
1.3	29 <sup>th</sup> June 2001	Shaun Sloan	Added Shop URL
1.4	15 <sup>th</sup> August 2001	Shaun Sloan	Reviewed and added screenshots

# Contents

<a href="#">1</a>	<a href="#">Scope of Document</a> .....	1
<a href="#">2</a>	<a href="#">CGIWrap - Overview</a> .....	2
<a href="#">3</a>	<a href="#">Web Site Administration</a> .....	3
<a href="#">3.1</a>	<a href="#">Virtual Hosts</a> .....	3
<a href="#">3.2</a>	<a href="#">Users</a> .....	4
<a href="#">4</a>	<a href="#">Directory Structure</a> .....	5
<a href="#">5</a>	<a href="#">Directory Permissions</a> .....	6
<a href="#">5.1</a>	<a href="#">Additional Sites</a> .....	7
<a href="#">6</a>	<a href="#">Apache Setup</a> .....	8
<a href="#">6.1</a>	<a href="#">ACCESS.CONF</a> .....	8
<a href="#">6.2</a>	<a href="#">HTTPD.CONF</a> .....	8
<a href="#">6.3</a>	<a href="#">SRM.CONF</a> .....	8
<a href="#">7</a>	<a href="#">Catalog - Network Settings</a> .....	9
<a href="#">8</a>	<a href="#">Shop URL</a> .....	9
	<a href="#">Appendix A</a> .....	10

## **1 Scope of Document**

The information in this document is intended as a guide to the setup of Cobalt RaQ Servers on which Actinic Host has been, or is to be, installed.

Once Actinic Host has been installed it is then intended to upload Actinic Business or Actinic Catalog.

## 2 CGIWrap - Overview

CGIWrap is a gateway program that allows general users to use CGI scripts and HTML forms without compromising the security of the http server. Scripts are run with the permissions of the "user" who owns the script. In addition, several security checks are performed on the script, which will not be executed if any checks fail.

CGIWrap is used via a URL in an HTML document. As distributed, CGIWrap is configured to run user scripts which are located in the ~/web/cgi-bin/ directory.

What this means is that the CGIWrap executable(s) live in the web server's default cgi-bin directory. It intercepts any CGI calls the default cgi-bin directory that include the word "cgiwrap" at the end and forwards them on to another directory that lives inside the user's "home" directory space. It also executes them "AS THAT USER". Hence, the files are owned by the "user" and not by the "web server user" (usually nobody). In this setup *nothing* is owned by the "nobody" user.

Cobalt RaQ server have CGIWrap installed by default.

### 3 Web Site Administration

Administration of a Cobalt RaQ Server is through a Web-based interface, that is username/password protected. Using this interface a new web site and user can be created and maintained. Because the administration is accomplished by this method it is not normally necessary to change any settings by accessing the server from the command line. Except for the creation of the cgi-bin directory – the web interface and Catalog create the rest.

#### 3.1 Virtual Hosts

Administration of Virtual Hosts on a Cobalt RaQ Server is accomplished through the Administrator web interface, and is consequently responsible for creating the necessary structures on the server.

As an example the VirtualHost of “ahost.actinic.com” has been created.

Set Virtual Site Defaults
Add Virtual Site

Virtual Site List			
? Host Name ↓	? IP Address ↕	?	?
ahost.actinic.com	10.1.40.25		
atest.actinic.com	10.1.40.25		
bttest.actinic.com	10.1.40.25		
testraq.design.com	10.1.40.25		

Details of site:




Site Settings	
? IP Address	<input type="text" value="10.1.40.25"/>
? Host Name	<input type="text" value="ahost"/>
? Domain Name	<input type="text" value="actinic.com"/>
? Bandwidth Limit	<input type="checkbox"/> On, allow <input type="text"/> kbps
? Accept Email for Domain	<input type="checkbox"/>
? Web Access by Domain	<input type="checkbox"/>
? Max. Allowed Disk Space (MB)	<input type="text" value="100"/> on the volume "home"
? Maximum Number of Users	<input type="text" value="50"/>
? Enable Shell Accounts	<input type="checkbox"/>
? Enable CGI Scripts	<input checked="" type="checkbox"/>
? Enable SSL	<input type="checkbox"/>
? Enable Server Side Includes	<input type="checkbox"/>
? Enable FrontPage Server Extensions	<input type="checkbox"/>
? Enable Secure POP3 (APOP)	<input type="checkbox"/>
? Suspend Site	<input type="checkbox"/>

**Note:** Ensure that “Enable CGI Scripts” is selected.

### 3.2 Users

Administration of users on a Cobalt RaQ Server is also accomplished through the Administrator web interface, and is consequently responsible for creating the necessary structures on the server.

As an example - User "Host Test" has been added with the "htest" User Name.

User List (1 - 1 of 1)			
Full Name	User Name	Email Aliases	
Host Test	htest	host.test	  

Details of user:

Modify User	
Full Name	<input type="text" value="Host Test"/>
User Name	<input type="text" value="htest"/>
Password	<input type="password"/>
Password (again)	<input type="password"/>
Max. Allowed Disk Space (MB)	<input type="text" value="50"/>
Telnet/Shell Access	Disabled
Site Administrator	<input type="checkbox"/>
Enable FrontPage User Web	Disabled
Secure POP3 (APOP)	Disabled
Suspend User	<input type="checkbox"/>

## 4 Directory Structure

The following section is for information purposes only, as explained above the administration of the server is accomplished through the Administrator web interface.

The directory structure for the first web site, as laid out on a Cobalt RaQ Server is normally as follows:

**/home/sites/home**

Below this directory the top level **users** directory:

**/home/sites/home/users**

Then the individual *username* directories:

**/home/sites/home/users/username1**

**/home/sites/home/users/username2**

The *username* in this example is the User Name – **htest** - created in [3.2](#) above:

**/home/sites/home/users/htest**

Below the *username* directory is the **web** directory:

**/home/sites/home/users/htest/web**

In the web directory are the following directories:

**/home/sites/home/users/htest/web/acatalog**

**/home/sites/home/users/htest/web/cgi-bin**

Plus the codebase tree

**/home/sites/home/users/htest/web/codebase/COM/Actinic/Catalog**

For any additional sites the format becomes:

**/home/sites/site1** instead of **/home/site/home** any additional sites will be **site2**, **site3** etc.

The System Administrator will need to create the following directories:

- acatalog
- cgi-bin
- codebase/COM/Actinic/Catalog

Then change the ownership of the directory to that of the user. For further details, see below.

## 5 Directory Permissions

The first web site **users** directory should be in the **home** group and be **suid**:

```
#pwd
/home/sites/home
# ls -l
drwxrwsr-x 17 nobody   home           1024 Jun 29 08:20 users
```

The **users** directory will be owned by the user that Apache is running under - in this example the "nobody" user.

In the **users** directory are owned by the User Name - **htest** - created in [3.2](#) above:

```
#pwd
/home/sites/home/users
#ls -l
drwxrws--x  3 htest    home           1024 Jun 27 17:29 htest
```

Additional user directories will also be created here:

```
drwxrws--x  3 username1  home           1024 Jun 27 17:32 username1
drwxrws--x  3 username2  home           1024 Jun 27 17:33 username2
```

In the individual **username** directories is the **web** directory that should also be owned by the **username**:

```
#pwd
/home/sites/home/users/htest
#ls -l
drwxrws--x  3 htest    home           1024 Aug 11 06:15 web
```

In the **web** directory are the **acatalog**, **cgi-bin** and **codebase** directories, which should also be owned by the **username**:

```
#pwd
/home/sites/home/users/htest/web
#ls -l
drwxrwsr-x  3 htest    home           3072 Jun 28 13:30 acatalog
drwxrwsr-x  2 htest    home           1024 Jun 28 13:17 cgi-bin
drwxrwsr-x  2 htest    home           1024 Jun 28 13:17 codebase
-rw-rw-r--  1 htest    home           5211 Jun 27 17:29 index.html
```

The **username** should also own the files within the **cgi-bin** directory:

```
-rw-r--r--  1 htest    home  13375 Mar 23 20:28 ad000001.pm
-rw-r--r--  1 htest    home  10858 Mar 23 20:28 ae000001.pm
-rw-r--r--  1 htest    home 224242 Mar 23 20:28 al000001.pm
-rw-r--r--  1 htest    home  175229 Mar 23 20:28 ao000001.pm
```

And so on.

The same applies to the **acatalog** and **codebase** directories.

The shops database files should be in the **web** directory and should be in the **username** group:

```
/home/sites/home/users/htest/web
-rw-r--r--  1 root    home    2 Aug 16 04:51 shops.dat.1
-rw-rw-r--  1 root    home    0 Aug 16 04:51 shops.dat.1.OPN
-rw-r--r--  1 root    home    0 Aug 16 04:51 shops.dat.1.inx
-rw-rw-r--  1 root    home    0 Aug 16 04:51 shops.dat.1.inx.OPN
-rw-r--r--  1 root    home  208 Aug 16 04:55 shops.dat.2
-rw-----  1 root    home    0 Aug 16 04:55 shops.dat.2.OPN
-rw-r--r--  1 root    home   84 Aug 16 04:55 shops.dat.2.inx
-rw-----  1 root    home    0 Aug 16 04:55 shops.dat.2.inx.OPN
```

## **5.1 Additional Sites**

The same structure and ownership is applied to any additional sites where the **/home/sites/home/users** would become **/home/sites/site1/users**, and this directory structure would have the ownership: **username site1**  
**/home/sites/site2/users** would be owned by **username site2** etc.

## 6 Apache Setup

The setup of Apache should be not normally need to be checked as the necessary entries/changes are accomplished by the Administrator web interface.

If it is necessary to check the Apache configuration the following files need to be checked for the relevant entries.

The entries may all be in httpd.conf or be spread across the three files listed.

### 6.1 ACCESS.CONF

Check the Options in the <Directory /home/sites> directive:

**Options Indexes FollowSymLinks Includes Multiviews**

Next, check the Options <Directory /home/sites/\*/> directive:

**Options -FollowSymlinks +SymLinksIfOwnerMatch**

### 6.2 HTTPD.CONF

In the VirtualHost directive check the following:

**RewriteEngine on**

**RewriteCond %{HTTP\_HOST} !^ipaddress(:80)?\$**

**RewriteCond %{HTTP\_HOST} !^dns\_name(:80)?\$**

**RewriteRule ^/(.\*) [http://dns\\_name/\\$1](http://dns_name/$1) [L,R]**

**RewriteOptions inherit**

**AliasMatch ^/~([^/]+)/(.\*)? /home/sites/home/users/\$1/web/\$3**

**AliasMatch ^/users/([^/]+)/(.\*)? /home/sites/home/users/\$1/web/\$3**

**AddHandler cgiwrapper .cgi**

**AddHandler cgiwrapper .pl**

### 6.3 SRM.CONF

Check the DocumentRoot:

**DocumentRoot /home/sites/home/web**

Check the ScriptAlias:

**ScriptAlias /cgiwrapDir/ /usr/cgiwrap/**

Check the Action:

**Action cgi-wrapper /cgiwrapDir/cgiwrap**

## 7 Catalog – Network Settings

There are no major changes in the network settings when uploading to RaQ server. So, in Advanced | Network Settings in Catalog/Business, the CGI-BIN URL will be: [http://web\\_site/~username/cgi-bin/](http://web_site/~username/cgi-bin/)

The codebase URL will be:

[http://web\\_site/~username/codebase/](http://web_site/~username/codebase/)

**Note:** The *~username* is the User Name – **htest** - created in [3.2](#) above.

ScriptID and Perl Extension would be as normal.

Following screenshot gives an example of the Advanced Network Setup:

**Advanced Network Setup**

**Configuration**

CGI Script ID Number: 408 Extension: .pl

Mail (SMTP) Server: mail.cix.co.uk

Web Site URL:

Catalog URL:

CGI-BIN URL: host.actinic.com/~hlest/cgi

Codebase: http://ahost.actinic.com/~h

Path From CGI-BIN To Catalog Directory: ../acatalog/

Path to the Perl shell: /usr/bin/perl

**FTP Details**

Server Host: ahost.design.com

Username: hlest

Password: [masked]

Path to CGI-BIN: web/cgi-bin/

Path from CGI-BIN to Catalog Directory as Viewed by the FTP Server (leave blank unless advised)

Ignore Passive Transfer Errors

Buttons: OK, Cancel, Apply, Proxy..., Convert..., Import..., Test

## 8 Shop URL

To browse to the shop URL is in the format:

[http://web\\_site/~username/acatalog](http://web_site/~username/acatalog)

**Note:** The *~username* is the User Name – **htest** - created in [3.2](#) above.

## Appendix A

An attempt to demystify the “regex” used in httpd.conf.

The VirtualHost directive contains the following lines:

### RewriteEngine on

This uses the *mod\_rewrite* capability of Apache, which allows URLs to be processed beyond all recognition, blocked conditionally, or even used to look up database records.

**RewriteCond** %{HTTP\_HOST} !^ipaddress(:80)?\$

Defines host IP address.

**RewriteCond** %{HTTP\_HOST} !^dns\_name(:80)?\$

Defines host DNS address.

Any number of RewriteConds directives can be prefixed to a RewriteRule to control its execution, and they are chained together.

**RewriteRule** ^/(.\*) [http://dns\\_name/\\$1](http://dns_name/$1) [L,R]

The RewriteRule is the core of *mod\_rewrite* and can have flags appended and conditional rules applied to it.

This rule is saying:

^/(.\*) “^/” is a start of line match for “/”.

(.\*) matches for any one character as many times as possible, and substitute it for \$1.

[L,R] “L” means it is the *last* rule to be processed. “R” means *redirect* and returns the new URL to the client.

### RewriteOptions inherit

This means that VirtualHosts and directories are to inherit the parent configuration.

**AliasMatch** ^/~([^/]+)/(.\*)? /home/sites/home/users/\$1/web/\$3

“^/~” start of match for “/~” “([^/]+) but no trailing “/” and substitute it for \$1.

The “/” of “/(.\*)?” is a match for a trailing “/” and would be \$2 if used. The “(.\*)?” Is a match for any other characters that are wholly optional, and would be substituted for \$3.

Some examples of what is achieved:

/foo/ fails \$1, \$2 and \$3 are empty

~foo fails \$1, \$2 and \$3 are empty

/~/ fails \$1, \$2 and \$3 are empty

~/foo succeeds \$1 is “foo” \$2 is “/” and \$3 is empty

~/bar succeeds \$1 is “bar” \$2 is “/” and \$3 is empty

~/foo// succeeds \$1 is “foo” \$2 is “//” and \$3 is “/”

~/foo/bar succeeds \$1 is “foo” \$2 is “/bar” and \$3 is “bar”

**AliasMatch** ^/users/([^/]+)/(.\*)? /home/sites/home/users/\$1/web/\$3

As for the previous AliasMatch, but looking for a match with “/users/” instead of “/~”.

### AddHandler cgiwrapper .cgi

Only if a file with specified extension will be treated as a CGI script.

### AddHandler cgiwrapper .pl

Only if a file with specified extension will be treated as a CGI script.