




Roxen WebServer 5.0

Administrator Manual

 Roxen Internet Software AB
© 2009 Roxen Internet Software AB. All
rights reserved.

Under the copyright laws, this document
may not be copied, in whole or in part,
without the written consent of Roxen
Internet Software.

Roxen Internet Software
Box 449
SE-581 05 Linköping
Sweden
www.roxen.com

Your rights to the software are governed by
the accompanying software license
agreement.

Every effort has been made to ensure that
the information in this document is
accurate. Roxen Internet Software is not
responsible for printing or clerical errors.

Other company and product names
mentioned herein are trademarks of their
respective companies. Roxen Internet
Software assumes no responsibility with
regard to the performance or use of these
products.

Contents

1	Roxen WebServer Administrator Manual	6
1.1	Introduction	6
1.1.1	Why Roxen?	6
1.1.2	Properties of Roxen	6
1.2	Installation	7
1.2.1	Installing the Software	7
1.2.2	Administration Interface	7
1.2.3	Sites	8
1.2.4	Migrating	8
1.3	Software Installation	8
1.3.1	Windows Binary Installation	8
1.3.2	Unix Binary Installation	10
1.3.3	Unix Source Installation	11
1.4	Creating a Simple Site	14
1.5	Creating a Roxen CMS Site	16
1.5.1	Creating the Site	17
1.5.2	SiteBuilder Storage	17
1.5.3	Access Control	18
1.5.4	FTP/WebDAV	19
1.5.5	Further Configuration	21
1.6	Upgrading Old Sites	21
1.7	Java Support	21
1.7.1	Servlets	21
1.7.2	Modules	22
1.8	Databases	22
1.8.1	Oracle	23
1.8.2	Sybase	24
1.8.3	Mysql	24
1.8.4	Msql	24
1.8.5	Postgres	24
1.8.6	ODBC	24
1.8.7	Informix	24
1.8.8	Mimer	24
1.9	Updating	25
1.9.1	Installing patches	25
1.9.2	Upgrading Roxen	25
1.9.3	Windows Binary Upgrade	25
1.9.4	Unix Binary Upgrade	25
1.9.5	Unix Source Upgrade	26
2	Administration Interface	27
2.1.1	Tab Overview	27
2.1.2	Administration Interface Site	28
2.2	Admin Tab	28
2.2.1	Your Settings	28
2.2.2	Users	30
2.3	Sites Tab	31
2.3.1	Site Page	32
2.3.2	Module Page	33
2.4	Globals Tab	34
2.5	Tasks Tab	34

2.6	DBs Tab	35
2.6.1	Database Groups	35
2.6.2	Databases	35
2.6.3	Creating New Databases	36
2.6.4	Database Buttons	36
2.6.5	Database Backups	36
2.6.6	MySQL Status	37
2.7	Docs Tab	37
2.8	Server Maintenance	37
2.8.1	Using the configure command	37
2.8.2	Using command line options	37
2.8.3	Options for ./start	38
2.9	Sites	39
2.9.1	Create a New Site	40
2.9.2	Delete a site	40
2.10	Site Information	40
2.10.1	URLs	41
2.10.2	Request Status	41
2.10.3	Cache Status	41
2.10.4	Events	41
2.11	Site Settings	41
2.11.1	Ports	42
2.11.2	Settings	42
2.11.3	Throttling	42
2.11.4	Cache	42
2.12	Logs	43
2.12.1	Event Log	44
2.12.2	Debug Log	45
2.12.3	Access Log	46
2.12.4	Accessed Counter	49
2.13	Globals	50
2.14	Port System	51
2.14.1	Ports tab	51
2.14.2	Protocols and the Internet	51
2.14.3	Numeric vs symbolic addresses	51
2.14.4	What it means for Roxen	52
2.15	Settings	52
2.15.1	UID	52
2.15.2	Threads	52
2.15.3	Modules Directories	53
2.15.4	Fonts	53
2.16	Auto Restart	53
2.16.1	Anti Block System	54
2.16.2	Auto Restart	54
2.17	Cache	55
2.18	Modules	55
2.18.1	Installing	56
2.18.2	Module Types	57
2.18.3	File Systems	59
2.19	Scripting	59
2.19.1	CGI	59
2.19.2	Fast CGI	60
2.19.3	SSI	60
2.19.4	Java servlets	60

2.19.5	Roxen modules	61
2.19.6	Pike scripts	61
2.19.7	Pike tag	61
2.19.8	Perl support	61
2.20	Performance: Average Profiling	62
2.20.1	Average Profiling Module	62
2.20.2	User Interface	63
3	Appendix A: Glob Expressions	65
4	Appendix B: Regular Expressions	66

1 Roxen WebServer Administrator Manual

1.1 Introduction

Welcome to Roxen CMS. In this chapter we will give an overview of the philosophy behind Roxen CMS's design. To make full use of Roxen it is helpful to know the ideas behind its design as well as how to use it.

1.1.1 Why Roxen?

The short answer is that Roxen tries to be a server that behaves like the web. Its user interface is accessed from a web browser from anywhere in the world, like a web site. Furthermore like the web Roxen is platform independent, it can run equally well on a number of operating systems. It is even possible to move a Roxen web site to another computer with no changes.

Another way to answer is to look at the internals of Roxen. Roxen CMS is highly modular; to do the simplest task Roxen needs at least a couple of modules. That might seem unnecessary at first, but it makes it possible to customize every aspect of the Roxen server.

Writing the actual modules is made simple because Roxen CMS handles complicated tasks such as the user interface. The module writer can concentrate on the actual functions in her module. This approach also ensures that the user interface works the same way, regardless of what new modules are invented.

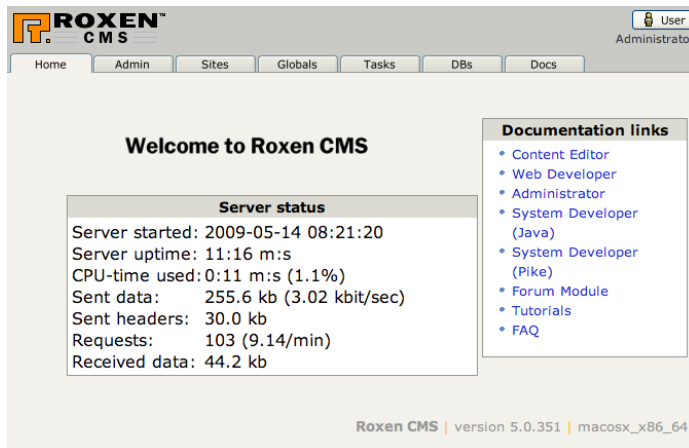
The best way to illustrate the advantages of the module interface is by taking a look at the FTP protocol module. Even the HTTP protocol handler is a module in Roxen. So by making a protocol module for FTP, Roxen became a FTP server as well. But it did not become just a plain FTP server since all scripting and database connectivity features are still available through FTP.

1.1.2 Properties of Roxen

Apart from the three important properties we have already mentioned, platform independence, a browser based administration interface and a modular architecture, Roxen is designed with scripting and database connectivity in mind. There are several ways to run scripts in Roxen. Standards such as CGI, FastCGI and Java Servlets are supported as well as the XML based macro language RXML.

RXML (Roxen Macro Language) is an easy-to-learn, easy-to-extend XML language. It makes it significantly easier to embed dynamic parts in HTML pages. Web designers can use the RXML tags themselves without having to learn programming, and programmers can create new RXML tags that can be reused on any number of pages with a minimum of effort.

1.2 Installation



Administration interface

To install Roxen CMS you first install the software, then connect to the web based administration interface and create the actual web or FTP sites. For each site you enable and configure the modules that are going to handle it. A basic web site only needs a small set of modules while a highly customized site that makes full usage of scripting and database needs more.

To make it simpler to get started there are predefined server templates that contain the necessary modules to get a certain type of web or ftp site up and running. By choosing a server template you do not have to manually enable those modules and you get a set of modules that gives you a working site.

If you already have a web site that you want to move to Roxen CMS this should not pose much of a problem. The toughest part is usually to get all scripts to run, since there may be slight differences between different implementations of standards such as CGI, SSI or .htaccess.

Some features relies on code outside of Roxen CMS to work. The Java support requires a Java 2.0 compliant virtual machine to be installed on the server. The database supports requires database connectivity libraries. Therefore it is harder to get these functions to work and it might be necessary to install software other than Roxen CMS itself.

1.2.1 Installing the Software

Roxen CMS consists of a Pike interpreter and Roxen CMS itself. The software is installed either from a binary or a source package. Binary packages are precompiled for each operating system while the user has to compile the server herself when installing from a source package. Source packages are available for Unix but not for Windows.

Usually the binary packages are a lot faster and easier to install. But some libraries might differ from your system in which case you are left with compiling a source distribution.

The last step of the software installation is setting a username and password and then starting the server so the administration interface becomes available.

1.2.2 Administration Interface

The bulk of the installation process is done through the web based administration interface. Here you create your sites and enable the modules you need. You can download additional software directly from the administration interface with the update client.

The administration interface is described in the Administration interface chapter.

1.2.3 Sites

Roxen CMS can handle any number of web or FTP sites with their own modules. A typical Roxen CMS installation uses at least two sites, one for the administration interface itself and another for the actual web site.

Each site is bound to at least one URL, through which it is possible to access the site. The URL is handled by a port within Roxen CMS. The ports are created automatically and can be further configured on the Ports tab. Several sites may share the same port. This is always done for IP-less HTTP, when several sites share the same IP number. But it is also possible for one web site to consist of several Roxen CMS sites. The administration interface could be given the URL `http://www.my-site/admin/` and appear as part of the web site `http://www.my-site.com/`.

How sites work is described in the Sites chapter while ports and URLs are described in the Ports chapter.

1.2.4 Migrating

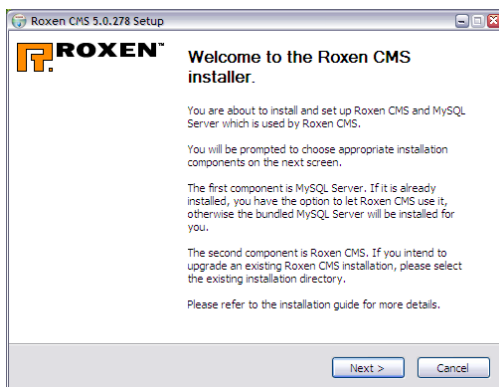
Moving an existing web site to Roxen CMS should pose no major problems, if Roxen CMS supports all scripts that are used on the site. Usually you only have to tell Roxen CMS where the web site is located. If scripts are used you also need to enable modules to handle the scripts. If your previous installation was using some special custom modules or similar that you for some reason can not run straight away under Roxen, a relaying solution is suggested.

1.3 Software Installation

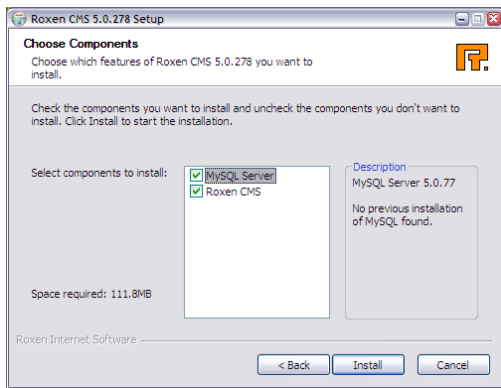
The software installation installs Pike and Roxen CMS on your system. It is either done through a binary or source package. Binary packages are precompiled for each operating system and are the easiest to install. Source packages are compiled by the user and require a complete development environment. They are for people who want full control over what they install, or a last resort if a binary package fails.

1.3.1 Windows Binary Installation

Start the Roxen CMS installation by double clicking the installation file. The installation wizard will guide you through the whole process of installing and configuring the software.



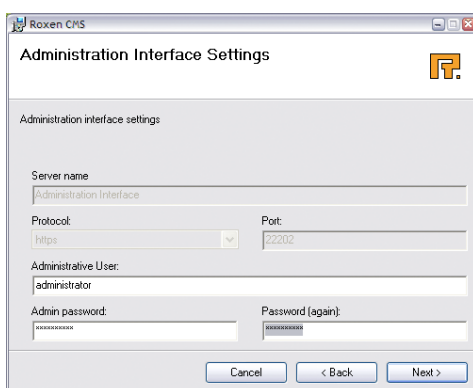
The installer lets you choose what components to install. If you already have a MySQL-server installed on your system, the installer will suggest the best course of action for you, based on the version of your current MySQL-installation.



If you have chosen to install MySQL the installer will extract the needed files and automatically start up the installer. Follow the steps on the screen to install the MySQL-server. If you intend to use MySQL solely together with the Roxen Webserver you can uncheck the box asking you if you want to configure MySQL before clicking *Finish*.



Next up is the CMS installer. The Roxen installer will once again automatically extract and start the installer for you. Follow the steps in the wizard and do not forget to enter a username and password for administrative access.



The last step will let you choose if you want to start the server right away. If you choose to you will be ready to access the web based administrator interface via a browser of your choice. Simply start your browser and point it to the port on the local machine that was specified during installation, typically 22202. Please refer to the following chapters for further instructions on how to set up your server.

1.3.2 Unix Binary Installation

The Unix binary release is distributed as a self-installable archive, actually a combination of a sh-script and a tar archive. The result is an executable with its own installation script that guides you through the installation.

During the installation process the install script will create and extract files to a temporary directory. It will create the directory in the current directory. The installation script will tidy up after itself, but if it is not able to it is safe to remove a directory with a temporary sounding name like DtmPf4a2b26/.

To install the binary release, simply run the archive:

```
sh roxen-[version].sh
```

Then press enter and accept the license text by answering yes.

It will take a while for the script to extract the Pike binary and start the real installation script. When it has finished it will prompt for the location of your installation:

```
Installation prefix ("roxen" will be added): /usr/local
```

If you do not enter anything Roxen CMS will be installed in /usr/local/roxen/. Roxen CMS is always installed in a directory called roxen. For automatic updating just enter the same location as the old installation. The old installation will be kept, but make sure the old server is not running. For more information see the Updating chapter.

Now the actual installation of the software will take place. This can take a while, depending on the server hardware. A progress bar will show the progress of the installation. Next the installation script will install Pike and proceed to create an administration interface server. If the installation updates an existing installation it will skip this step.

The script will prompt for the name and URL of the administration interface server, as well as the user name and password for an administration interface account.

Note!

It is no longer necessary for the administration interface to run under its own port, you could assign it a URL on your normal server, for example <http://my-domain/admin/>. See the Ports chapter for more information.

The https protocol should be used for the administration interface, otherwise it might be possible to eavesdrop on it. To make it secure it is not enough just to choose a https URL, it is also necessary to create a new certificate, since it will use the insecure demo certificate by default.

```
Creating an administration interface server in
/usr/local/roxen/configurations/.
```

```
Server name: Administration Interface
Port URL: http://*:22202/
Administrator user name: administrator
Administrator password:
Administrator password (again):
```

```
Are the settings above correct [Y/n]?
```

With this the installation script has all the information it needs. It will finish the installation and start the Roxen CMS server. Further configuration is done by connecting to the administration interface, the installation script prints the URL. If it is not possible to connect to the administration interface, the debug log file should give clues to what went wrong. It is located at roxen/logs/debug/default.1.

```
Administration interface created.

Administrator user "administrator" created.

Would you like to start the server now [Y/n]?

The initial installation is complete. Starting Roxen,
please wait...Updated nsr file "../logs/.nsr"
Updated nsr file "../logs/debug/.nsr"
...

Congratulations! Roxen has started. You may now proceed with the
installation from the administration interface using a web browser:

http://holt:22202/
```

1.3.3 Unix Source Installation

The Unix source release is distributed as a tar.gz archive. It contains the source code for the Pike interpreter, necessary libraries and the Roxen CMS server itself. The installation will work on any system, but libraries for all supported functionality is not included. Thus libraries for native database support, for image file format support (TIFF, JPEG), TrueType rendering, Java, or GTK/GNOME support are not included.

If you are installing on a Linux or FreeBSD system most libraries will be shipped with the distribution. The commercial Unix system does however usually not ship these libraries, so you will have to install them yourself.

It is necessary to install the external libraries before you install Roxen CMS. If you later find out that some function is lacking because of a missing library you will have to reinstall Roxen CMS, or rather the Pike interpreter. A reinstallation is done in the same way as an installation and does not affect the web sites you have created. The installation script will find the previous installation of Roxen CMS and upgrade rather than overwrite it.

The installation process consists of four steps; configuration, compilation, software installation and Roxen CMS installation. For detailed options using configure, see the Maintenance chapter for details.

```
./configure --prefix=/usr/local
make
make install
cd /usr/local/roxen/server
./create_configinterface
```

The configuration is done automatically by a script created with GNU autoconf. It finds out about your system and how to compile on it. It will find which libraries are installed and if they can be used. Usually this will be fully automatic but in some cases it might be necessary to manually force the configuration system to find some libraries. Use ./configure -help for a list of configurations options.

```
./configure --prefix=/usr/local/
```

The configure script produces a log about what happens. A longer version of the log is also stored as config.log. If the configure script failed to find a certain library that does exist the config.log file will give you valuable clues to why, and what you can do to fix it. Once you have fixed the problem you can run configure again.

```
holt roxen-2.1.130-src $ ./configure --prefix=/usr/local/
loading cache ./config.cache
checking whether make sets ${MAKE}... (cached) yes
checking for a BSD compatible install... (cached) /usr/local/bin/install -c
configure: warning:  Converted . to /home/wing/cto/manual/roxen-2.1.130-src,
If this does not work, please use an absolute path to the configure script.
creating ./config.status
creating Makefile
configuring in pike/src
running /bin/sh /home/wing/cto/manual/roxen-2.1.130-src/pike/src/configure --
prefix=/usr/local/
--cache-file=../../config.cache --srcdir=/home/wing/cto/manual/roxen-
2.1.130-src/pike/src
loading cache ../../config.cache
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
```

Beginning of configure log

After the software is configured, it is time to compile it. This is done by invoking make which will in turn use the necessary compilers, linkers and whatnot. The compilation takes a while and produces another long log of information. Usually all messages can be ignored, but in case of errors they will provide important clues to what went wrong.

make

```
holt roxen-2.1.130-src $ make
make[1]: Entering directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src'
Creating lib/.
Creating lib/modules/.
Creating lib/include/.
Creating symlink ./share
Done.
t="$PIKE_PATH_TRANSLATE"; if test "x$t" = "x"; then t=s,x,x,; else ;; fi; \
sed -e 's!$lib_prefix!'"`echo \"/home/wing/cto/manual/roxen-2.1.130-
src/pike/src/lib\" | sed -e
\"$t\"`"'! </home/wing/cto/manual/roxen-2.1.130-src/pike/lib/master.pike.in \
| sed -e 's!$share_prefix!'"`echo \"/home/wing/cto/manual/roxen-2.1.130-
src/pike/lib\" | sed -e \"$t\"`"'! >master.pike
```

Beginning of make log

When the software has been compiled it is time to install it. This is usually done by make install. The make command, that is used to handle the compilation process, is used to handle the installation process as well. The software will be installed in the directory that you specified with the --prefix= argument to configure, by default /usr/local/.

Roxen CMS and Pike will both be installed separately, in their own directories (pike/ and roxen/). A link to the pike binary will also be created in bin/. The roxen directory can later be moved anywhere on your system. However the pike directory cannot be moved. In case you want Pike

and Roxen CMS in different places you should set the `--prefix=` argument to the configure script for Pike, not Roxen CMS.

Upgrading to a newer release, or reinstalling the same release, is also done with `make install`. Only the `roxen/server/` directory, that contains the actual software, will be affected. No logs, configurations or user data will be touched. After the installation the old version of the software will be moved from `roxen/server/` to `roxen/server.old/`.

The installation also produces a progress log that is mainly interesting in case of errors.

```
make install

holt roxen-2.1.130-src $ make install
make[1]: Entering directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src'
make[2]: Entering directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src/modules'
Making CommonLog
make[3]: Entering directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src/modules/CommonLog'
make[3]: Leaving directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src/modules/CommonLog'
Making Gdbm
make[3]: Entering directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src/modules/Gdbm'
make[3]: Leaving directory `/home/wing/cto/manual/roxen-2.1.130-src/pike/src/modules/Gdbm'
```

Beginning of make install log

The next step is to create an administration interface server. If you are upgrading or reinstalling you do of course not need to, your old configuration will do. This is done by invoking the `create_configinterface` script located in the `roxen/server/` directory. If is the same script that is used by the binary installation, so we won't describe it in detail again.

```
cd /usr/local/roxen/server/
./create_configinterface

    Creating an administration interface server in
    /usr/local/manual/roxen/configurations/.

Server name: Administration Interface
Port URL: http://*:22202/
Administrator user name: administrator
Administrator password:
Administrator password (again):

Are the settings above correct [Y/n]?

    Administration interface created.

    Administrator user "administrator" created.
```

The last step of the software installation is to start the server and connect to the administration interface. This is done by invoking the start script in the `roxen/server/` directory. The progress of the server can be seen in the debug log, `roxen/logs/debug/default.1`

```

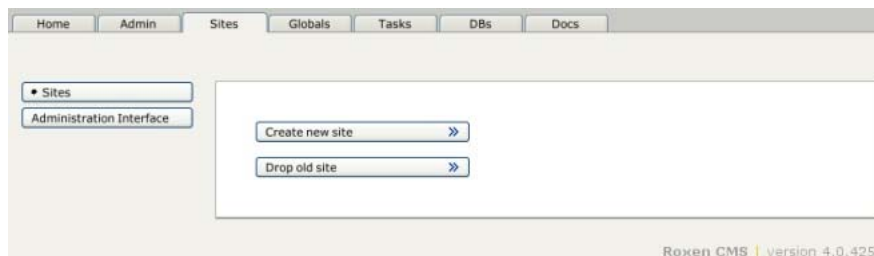
cd /usr/local/roxen/server/
./start

7:54:46 : -----
0m 0.0s : Pike v7.0 release 230, Roxen WebServer 2.1.130
      : Loading pike modules ... Done [855.0ms]
      : Loading roxen ... Done [7884.8ms]
      : Loading font handlers ...
      :   Builtin fonts (scalable)
7:54:55 :   Compact image file font (bitmap)
0m 9.5s :   Image directory fonts (bitmap)
      :   Image TAR-file fonts (bitmap)
      :   Compatibility bitmap fonts (bitmap)
      :   TTF fonts (scalable)
      : Done [516.5ms]
7:54:56 : Adding languages ... Done [45.05ms]
0m 9.7s : Initiating argument cache ... Done [101.31ms]
      :
      : Enabling the configuration Administration Interface ...
      : Registered http://*:27474/ for Administration Interface
      : Enabled Administration Interface in 412.1ms
7:54:56 : Starting 5 threads to handle requests.
0m10.2s : -- Total boot time 10.2 seconds -----
      :
      : ** 2000-09-20 07:54   pid: 15469   ppid: 15460   uid: wing

```

Beginning of debug log

1.4 Creating a Simple Site



Sites tab

A site is created by choosing the Create new site button under the Sites tab.

Create new site

Site name:

The name of the configuration must contain characters other than space and tab, it should not end with ~, and it must not be 'CVS', 'Global Variables' or 'global variables', nor the name of an existing configuration, and the character '/' cannot be used. This name will show up in the sites menu, a lowercased, filesystem-friendly version will become the virtual server's configuration file name as well as the name of its log file directory.

Create new site

The first step is to give a name to the site. The name is only used to identify the site in the administration interface, it will not be exposed externally.

Press Create new site with template to continue.

Choose Standard site.

Initial variables for the site

URLs

Protocol	Host	Port	Bind this port
http	/*	80	Yes

New row

Bind to these URLs. You can use '*' and '?' to perform globbing (using any of these will default to binding to all IP-numbers on your machine). If you specify a IP# in the field it will take precedence over the hostname.

Initial variables for Directory Listings

Index files

index.html	▲ ▼ Delete
index.xml	▲ ▼ Delete
index.htm	▲ ▼ Delete
index.pile	▲ ▼ Delete
index.cgi	▲ ▼ Delete

New row

If one of these files is present in a directory, it will be returned instead of the directory listing.

Initial variables for File systems: Normal File system

Mount point

Where the module will be mounted in the site's virtual file system.

Search path

NONE

The directory that contains the files.

OK

Initial variables

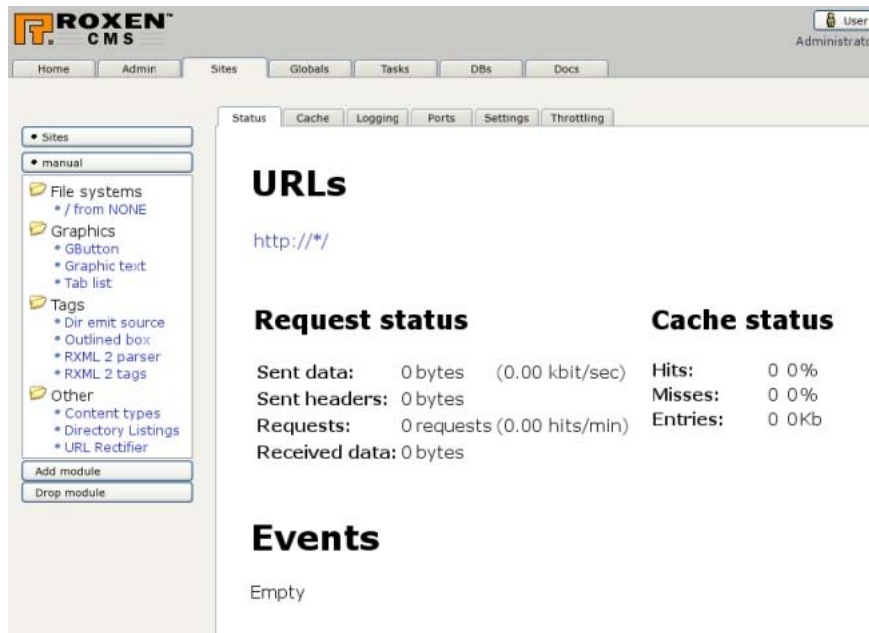
Now you chose which URL the site is to have. An asterisk (*) instead of a host name means that it will answer to any domain name that is handled by the computer. You should usually give a domain name instead; http://your-domain/. You do not have to specify the port number explicitly.

Roxen handles several additional protocols: FTP and HTTPS most notably.

If you want to use another protocol you just change the URL accordingly, e.g ftp://your-domain/ or https://your-domain/. These ports almost always need to be configured in some way. A HTTPS port needs a certificate, a FTP port uses different authentication, etc. Configuration is done under the Ports tab, after the site has been set up. See the Ports chapter for more information about the ports system.

The last step is to configure required variables in modules. For the standard site template you only need to configure the Search path variable in the File system module. This is the directory where your web site is stored, and corresponds to the Document root setting of other web servers.

Once you have saved the File system module settings the site is up and running. It will contain some basic modules as well as modules for RXML parsing. You probably want to make some adjustments to the installed modules as well as add more modules for your specific needs.

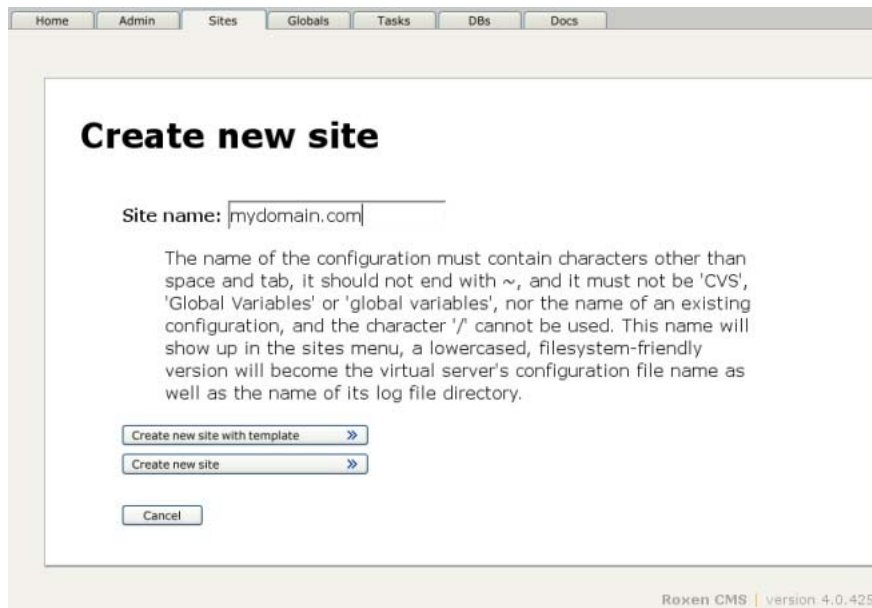


Finished installing

To configure a module, choose it in the module list to the left. To add a module, press the Add Module button. The rest of this chapter gives ideas about how to install modules that provide a specific function, such as scripting or database access. The rest of the manual goes into depth about all modules, and what configuration options are available.

1.5 Creating a Roxen CMS Site

The main job of setting up a CMS Advanced site is determining how your organization should make use of the access control and work flow capabilities. We will however not go through that here, but rather in the SiteBuilder chapter. Here, we cover how to set up a simple single server CMS Advanced installation; more details on additional steps to set up a multiserver setup is covered in the multiserver section.



Naming your sites after the domain they bind could prove a helpful choice.

1.5.1 Creating the Site

As with a standard site, a CMS Advanced site is created with the Create new site button under the Sites tab. But after naming the site and choosing Create site with template, select Roxen CMS site instead of Standard site.

The question of a URL is the same as for a standard site. On top of that, you are prompted for a SiteBuilder storage directory.

Note!

If you are setting up the server on an NT system, the search engine requires either that the first server URL you give the site is a fully qualified hostname (i e www.roxen.com, instead of just www or *) or that your NT systems primary DNS suffix is set correctly! Currently you also need to add at least one non-https port for the indexer.

1.5.2 SiteBuilder Storage

The SiteBuilder storage is a directory that contains all information about a SiteBuilder site, including the CVS repository where the files are stored, configuration information and dumps of the access control database. The directory can be copied anywhere, in case the site is to be moved to another machine or for backup purposes, if you dump the AC database (this is done under the Access Control tab in the content editor, sub tab Maintenance).

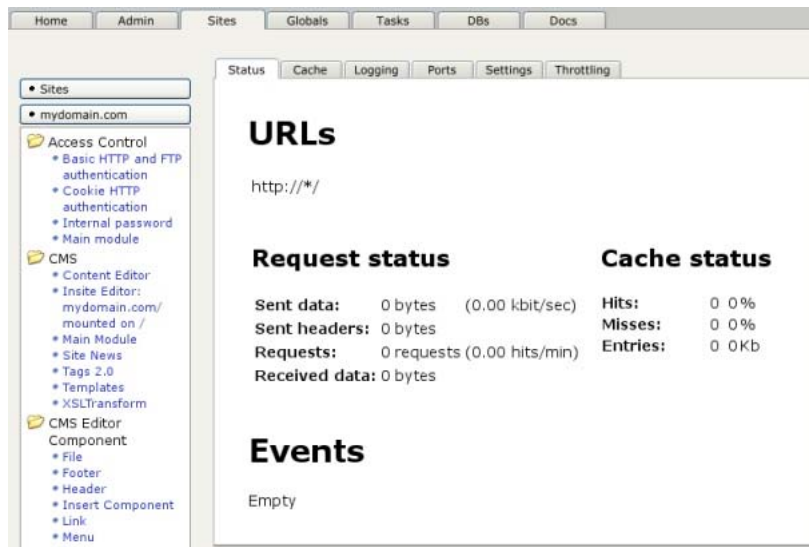
Note!

It is not possible for several SiteBuilder to use the same storage.

To upgrade from a site from a previous release, simply enter the path to the old storage, or to a copy of it. Creating a new storage is just as simple. Enter a path to a non-existing directory, or keep the suggested default path. This choice can also be redone at a later time, should you choose to switch SiteBuilder storage for the site; just change the setting in the SiteBuilder: Main module module and reload it to set up the new storage.

Next, you are prompted with whether you want a basic site, a demolabs site or if you want to start from an empty site with just an administrator user setup. Regardless of your choice, you

will then be prompted for the name and password of the initial administrator user and, if you chose demolabs, the example users.

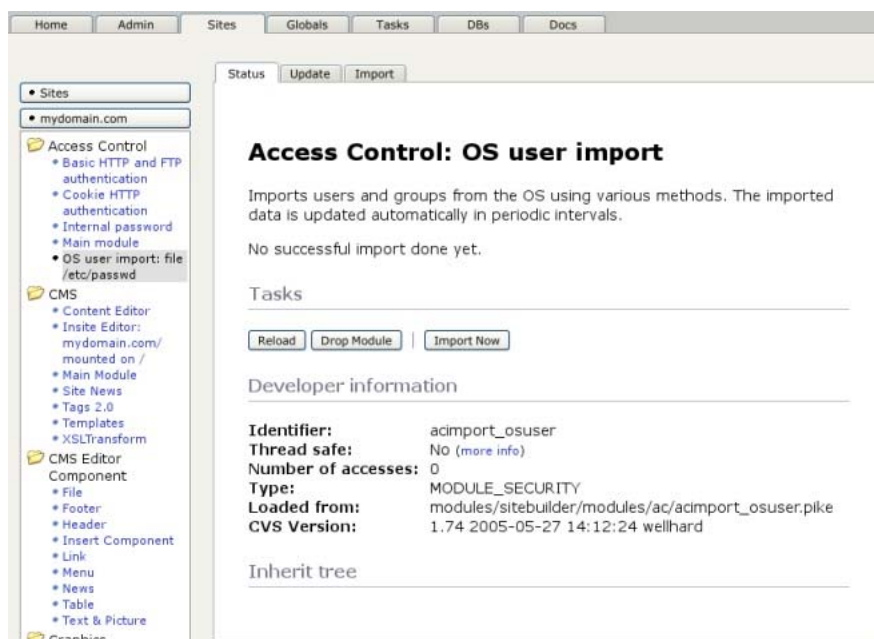


Installation complete! This, or something like it, is the site you just created.

1.5.3 Access Control

Now that the site is up and you have an administrator user, it is time to create users for all persons who are going to work with the web site. SiteBuilder is designed so that everybody should have a personal account. Only then will it be possible to track changes in the versioning repository or make full use of the access control system. Once everybody has a user, and has been assigned the proper rights, it is unnecessary to keep the special Administrator user.

There are several ways to get users into the access control database. The simplest one is to import accounts from another system. That way, the users won't have to keep track of another password, nor do you have to administrate accounts in two systems. Importing accounts is done using the module OS user import or the LDAP user import module.



OS user import module

Manually setting up passwords is done under Access Control tab, sub tab Identities in the content editor (which by default is located at <http://your-domain/edit/>).

The screenshot shows the 'Identities' tab in the Roxen WebServer Administrator. The 'User: Administrator' is selected. Under 'Details', the 'Name' is 'Administrator', 'Username' is 'administrator', 'Zone' is 'Common', and 'Internal ID' is '2'. There are 'Save' and 'Delete' buttons. Under 'Authentication methods', 'Access Control: Internal password' is selected and enabled. It has fields for 'Password', 'Again', and 'Encrypted: Pvd/csMeQUhT'. There are 'Save' and 'Delete' buttons. At the bottom, there is a section 'Add a new authentication method:' with a dropdown menu showing 'Access Control: Internal password' and an 'Add' button.

Changing the Administrator password

1.5.4 FTP/WebDAV

Another way to edit a SiteBuilder site is through FTP/WebDAV. This is nice when using editors that have built-in FTP support, as well as for uploading and downloading a many files. To enable FTP it is necessary to create a new site. A FTP site can handle several SiteBuilder sites. This is because each FTP site needs its own IP number, something that is not necessary for HTTP sites.

The screenshot shows the 'Sites' tab in the Roxen WebServer Administrator. The 'License file' is set to 'None'. Below this, the 'Roxen CMS' section lists several site templates: 'CMS site', 'CMS Content Editor Frontend site', 'CMS Replication Frontend site', and 'CMS FTP/WebDAV edit'. The 'CMS FTP/WebDAV edit' template is highlighted with a blue bar and a double arrow icon. Each template has a brief description below it.

Create new Site

A FTP site is created by choosing Create new site with template under the Sites tab, naming it and then choosing the CMS FTP/WebDAV edit site template.

Initial variables for the site

URLs

ftp	://*	:
21		
IP#:		Bind this port:
Yes		

New row

Bind to these URLs. You can use '*' and '?' to perform globbing (using any of these will default to binding to all IP-numbers on your machine). If you specify a IP# in the field it will take precedence over the hostname.

SiteBuilder mydomain.com

The SiteBuilder whose work areas this filesystem should access.

OK

URL

As always you need to configure the URL of the server, which should be of course be a FTP URL.

Home Admin Sites Globals Tasks DBs Docs

Status Settings Security

• Sites

- ftp.mydomain.com
- Access Control
 - User Database
- CMS
 - Unassigned filesystem
 - XSLTransform
- WebDAV
 - Protocol support

Add module

Drop module

Location

The location of the work area filesystem. Note that setting this to anything other than '/' for a filesystem in view mode will probably break the site, unless relative paths are used in all links.

Work area

None

The work area, in any running SiteBuilder on this server, that this filesystem should access.

Language

ANY

Restrict access to multi-language files to the given language.

Download mode

Edit

Restore default value (View)

Save

CMS: Work Area Filesystem

The CMS: Work Area Filesystem module needs configuring. First choose one of the edit modes, "Edit", "All work areas" or "All sites", for the Download mode variable. Unlike the view mode the edit modes send the original files rather than parsed files. The "Edit" mode is for editing one work area, the "All work areas" for editing all work areas on one SiteBuilder site and the "All sites" mode for editing all sites on this Roxen server.

After choosing the correct download mode and saving that settings it is necessary to reload the page. For the "Edit" mode it is now necessary to choose which work area should. For the "All work areas" mode it is instead necessary to choose for which SiteBuilder server.

Home Admin Sites Globals Tasks DBs Docs

Status Settings

• Sites

- ftp.mydomain.com
- Access Control
 - User Database
- CMS
 - Edit: mydomain.com: All work areas mounted on /
 - XSLTransform
- WebDAV
 - Protocol support

Add module

Drop module

AC database

mydomain.com

Restore default value (None)

The server containing the AC database to use. The 'all' alternative authenticates against all databases in all virtual servers.

Save

AC: User Database

Once the Work Area Filesystem module is configured the AC: User Database needs to be configured. The configurable setting is the AC database, which should point to the SiteBuilder site that handles the work area(s), unless the download mode is "All sites" in in which case it should be "All".

Finally it is probably necessary to change a few settings in the FTP port used. It is found under the Ports tab, unless an error occurred while trying to open the port. Usually the Allow anonymous ftp and Allow login with incorrect password/user should be set to "No".

1.5.5 Further Configuration

Now you should have a CMS Advanced site that is up and running and which can be edited by the tools of your choice. However this does not mean that everything is configured. There are still plenty of things to configure:

- Access Control groups and permissions
- Workflow paths
- Work areas
- File types
- Languages

See the SiteBuilder chapter for details.

1.6 Upgrading Old Sites

When upgrading your Roxen installation it is important to make sure your sites are up to date in order for you to enjoy the full benefits of your upgraded server.

On our website you can find up to date information on how to do this for your current setup. Please visit <http://www.roxen.com/support/>.

1.7 Java Support

Roxen's Java support requires a Java 2.0 (or 1.2 since it is the same thing) compliant virtual machine. The binary distributions have been compiled against Sun's JVM, to use the Java support it is necessary to install Sun JRE or JDK 1.2.

To compile the Java support into a source release of Roxen it is necessary to have the full JDK installed.

If you have Java support loaded the Java servlet bridge module will appear among scripting modules, and the Java example module will be shown among the example modules.

1.7.1 Servlets

A Java servlet is installed by adding a copy of the Java Servlet bridge module. The servlet's location is configured in the Code directory variable, either as a directory or a jar file. The servlet itself is identified by the Class name variable. Each servlet gets its own location in Roxen's virtual file system, which is configured with the Servlet location variable.

File extension servlet

Use a servlet mapping based on file extension rather than path location.

Servlet location

This is where the servlet will be inserted in the namespace of your server.

Class path

Any number of directories and/or JAR files from which to load the servlet and its support classes.

Class name

The name of the servlet class to use.

Parameters

Parameters for the servlet on the form *name=value*, one per line.

Parse RXML in servlet output

If this is set, the output from the servlet handled by this module will be RXML parsed. NOTE: No data will be returned to the client until the output is fully parsed.

Priority

The priority of the module. 9 is highest and 0 is lowest. Modules with the same priority can be assumed to be called in random order.

Java servlet bridge settings

Parameters specified as *name=value* in the Parameters variable is sent to the servlet. The servlet can use RXML tags in its result if the Parse RXML in servlet output variable is enabled.

Once you have installed the servlet the modules information tab will show information about that module, or an error message in case the servlet could not be loaded.

1.7.2 Modules

Java modules are just .class files or .jar files within the normal module directories. The example module is an example in the examples section. Java modules work exactly the same way as Pike modules.

1.8 Databases

Roxen contains a generic database API, which is used to connect to a number of different SQL databases, through either a native or an ODBC driver. The native drivers are compiled into Pike.

Each database is then accessed through a database URL with the format:
 driver://user:password@host[:port]/database. For example mysql://test:Test
 password@mysqlhost.roxen.com/test.

To check the database connectivity use the SQL tag module located in the database section, and configure a database path.

The screenshot shows the 'Security' tab of the Roxen WebServer Administrator. On the left is a sidebar with a tree view containing categories like 'Sites', 'Access Control', 'CMS', 'CMS Editor Component', and 'Graphics'. The main content area has tabs for 'Status', 'Settings', 'Security' (selected), and 'Documentation'. Below the tabs are several configuration sections:

- Default database:** A dropdown menu set to 'sb_test'. Below it, text explains that if both 'Allow SQL URLs' and 'Allowed databases' are disabled, this is the only database the tags will use.
- Allow SQL URLs:** A dropdown menu set to 'No'. Text explains that allowing generic SQL URLs can be a security hazard.
- Allowed databases:** An empty text input field. Text explains that this is a comma-separated list of databases allowed in the 'db' attribute.
- Support "module" attribute:** A dropdown menu set to 'No'. Text explains that this supports the deprecated 'module' attribute.
- Default charset:** An empty text input field. Text explains that this is the default value for the 'charset' attribute.
- Priority:** A dropdown menu set to '5'. Text explains that priority ranges from 9 (highest) to 0 (lowest).

Each section has a 'Save' button at the bottom right of the main content area.

SQL tag module

If everything works the information tab will give information about which database you are connected to.

If anything does not work the information tab will give an error message. For example if the server could not find the correct driver. Or if it could connect to the database server, but the user name, password or database was wrong.

The database support can be used for many things within Roxen. The most common usage is to use RXML tags to produce output from a database. But other modules use the database connectivity as well. Usually you enter the whole database URL when you configure modules. But you don't really want to do that in each web page. Therefore the SQL Databases module can be used to give symbolic names to database URLs. In the pages it is only necessary to use the symbolic name. If the database moves it won't be necessary to change any pages.

To find out which database drivers are available in your Pike, use the Pike module list task under the Tasks tab. The available database drivers will be shown under Features.

1.8.1 Oracle

Oracle 7 and 8 are supported through a native driver. This requires that the Oracle client libraries (OCI) are installed. (If installing Roxen from the source distribution, Oracle and OCI must be installed before Roxen.)

URL format: `oracle://user:password@SID/`

The SID is the Oracle Server ID, and may be omitted to access the default server. Optionally, a tablespace identified may be added after the final slash.

1.8.2 Sybase

Sybase is supported through a native driver.

1.8.3 Mysql

Native support, mysql://user:password@host:port/database. If the driver is compiled into your Pike you don't need to do anything further.

1.8.4 Msqldb

Native support, msqldb://user:password@host:port/database.

1.8.5 Postgres

Native support, postgres://user:password@host:port/database.

1.8.6 ODBC

ODBC is configured through a .odbc.ini file. The file is either found in the Roxen user's home directory or with the ODBCINI environment variable. The .odbc.ini file can contain configurations for several databases. The syntax is as follows:

```
[name]
Driver = path
option = value
...
```

Name is a name set, that will be used to refer to this database. Driver is the path to the ODBC driver, in the form of a dynamic library. Option is various options that will be forwarded to the driver. Available options vary between drivers. Usual options include the network location of the database manager as well as the name and password of the database user.

```
[toronto_wp]
# white paper of metro Toronto
Driver = /usr/lib/odbc/oracle.so
<...>

[netnews]
# NNTP netnews group
Driver = /usr/lib/odbc/nodbc.so
Server = news.empress.com

[rnd_test]
# data source for R&D test
Driver = /home/r_d/odbc/empodbc.so
URL = empodbc://rnd.empress.com:6322/rnd_test/testdb
Sample .odbc.ini file
```

The database URL for a ODBC database differs because you specify the name given to the database instead of the host name and port. Thus an odbc database URL looks like this: odbc://username:password@name/database.

1.8.7 Informix

Informix is supported through a native driver.

1.8.8 Mimer

Mimer is supported through its ODBC driver.

1.9 Updating

1.9.1 Installing patches

Applying patches can be done either from within Roxen's Administration interface or by using the command line tool that can be accessed through the terminal (UNIX) or the command shell (Windows). Applying patches is done in two stages. The first stage is called Import. This will extract all the files provided in the patch package and make them accessible to Roxen. You can list imported patches, see what files will be affected when applying the patch and read the patch description once the patch has been imported. If you uninstall any patch later it will go back to being just imported. To remove a patch completely you need to manually delete its folder from the import folder in your Roxen installation.

Note!

When uninstalling a patch all files affected by the patch will be returned to the state they had when they were patched. Any customisations made to those files after the patch was installed will be lost.

1.9.2 Upgrading Roxen

Manually upgrading your Roxen is in most cases very similar to installing it. A new server-x.x.x directory will be installed where x.x.x is the Roxen version including a build number. The old server directory will be kept. You can manually delete this when you are satisfied with the upgrade.

Always when upgrading a Roxen installation, remember shut down the server during the update session to avoid any errors from occurring.

1.9.3 Windows Binary Upgrade

Upgrading a Windows installation follows the same procedure as for the Windows installation. Within the wizard, specify the current location of the installation. Roxen will now install a new server directory in the same fashion as for Unix, see below. Configurations files and files created by others than the installation program will not be overwritten. However, be advised to make copies of any Roxen file that you have modified to prevent accidental information loss.

1.9.4 Unix Binary Upgrade

The installation script takes a number of options; use `-help` to list them. For instance, `-without-start` makes sure Roxen will not start after the update is complete.

```
sh roxen-[version].sh [--options]
```

It will take a while for the script to extract the Pike binary and start the real installation script. When it has finished it will prompt for the location of your current installation:

```
Installation prefix ("roxen" will be added): /usr/local/
```

Enter the path to your current Roxen installation. Remember not to add "roxen" to the path since the installation script itself adds "roxen" to the path. The script will install the new version of Roxen in `roxen/server-x.x.x/`, where x.x.x is the Roxen version including a build number.

Take care not to keep any essential code in your old `roxen/server/` directories. After three upgrades Roxen begins deleting old server directories. Keep vital code in your `roxen/local/modules/` directory instead. After each update, make sure any local changes in the old Roxen-source is moved to the new `roxen/server/` directory.

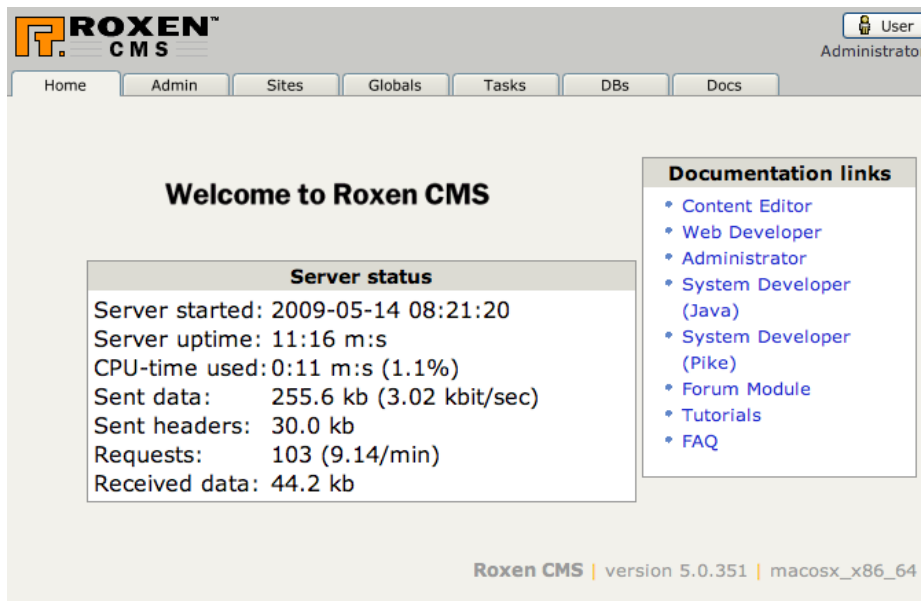
1.9.5 Unix Source Upgrade

The result when upgrading a Roxen installation compiled from the source is the same as when updating a binary installed Roxen. What differs however is the procedure.

```
./configure --prefix=/usr/local  
make  
make install
```

Follow the instructions in the installation manual but only through the first three commands as you already have a configurations interface available.

2 Administration Interface



Administration interface

Configuration and maintenance of Roxen is handled by the web-based administration interface. The administrators connects to the interface with a browser and can then access all server-related configuration options and maintenance tasks. You can set up multiple administration interface users with varying privilege levels and their own preferred look and feel to the interface.

2.1.1 Tab Overview

The administration interface is split into seven tabs, all covering a specific subset of server administration or information features. A tab will remember what subpage you visited the last time you accessed it, and recall this page the next time you switch back to that tab.

Home

The intro screen you arrive at when accessing the initial page of the admin interface - a page where you may choose to receive regular updates of the status of Roxen bugtracking tickets, the most recent articles written for Roxen Community, a brief run-down of the server's status or news from www.roxen.com.

Admin

The Admin tab contains settings for the administration interface itself.

Sites

The Sites tab contains the configurations of the web and ftp sites handled by the Roxen server. Most of the administration work is controlled from here - creating sites, adding or dropping modules to them or configuring their operating parameters.

Globals

The Globals tab contains settings and information relevant to the whole Roxen server, such as where to find modules or which UID to run under. The global event log resides here; a web based version of the debug log. All error messages will show up here, as well as in the local event logs of the site and module instance that produced them.

You'll also find a summary view of all ports that are open for all sites handled by the server.

Tasks

The Tasks tab contains support wizards that are not related to configuring the server but some other task, such as restarting the server, flushing memory caches, creating SSL certificates or getting debug information.

DBs

This is where you register, browse and peek at the databases and tables in or related to your Roxen installation - you may access any database from here, not just the internal MySQL one. This browser both has a click-to-peek interface, and the more powerful SQL prompt for digging out any and all data you have access to. You may even look at images stored in, for instance, the gtext_data table of the local database.

Docs

The Docs tab contains the online documentation. The latest version of the documentation for Roxen WebServer, the free part of CMS Advanced, can be found at docs.roxen.com.

2.1.2 Administration Interface Site

The administration interface is in fact a site just like any other web site handled by Roxen. It is implemented using RXML and a few custom modules - but since it is the average administrator's only way of handling the server, it is by default locked, so that you can not remove, add or alter its operating parameters, apart from what URLs it should reside on.

The administration interface site should use https, to get a secure connection. It is however necessary to create a new certificate to make this secure, since it will use the demo certificate shipped with Roxen by default. Since anyone who downloads Roxen can get that certificate, it is not secure.

2.2 Admin Tab

Below the Admin tab you have access to the Update server, your personal settings for the look and feel of the administration interface, and the user database for the administration interface, where you set up what users have what administrator rights.

2.2.1 Your Settings

Under this subtab, you may configure the admin interface to your personal preferences. Any and all settings made here affect only the user you are currently logged-in as, "administrator" if you didn't change the name suggested in the installation phase. This list intentionally does not cover all options, but focuses on a selection of the more interesting ones.

Content Boxes

The content boxes let you put various info or news on the start page at the root of the admin interface; alerts of new articles at Roxen Community, info on recently posted tickets to the Roxen Internet Software bugtracking system BugCrunch, server status or other things. Try them out!

Theme

The single most visible configurable option of the admin interface is the theme selection, where you choose whether you want the most recent brand of layout from the Roxen marketing department, the lovely colorful touch of the Power Puff Girls, or

something plain different. If you administer several Roxen servers and want to differentiate them from one another, set up your administrator user to different themes for different servers.

With a little tweaking of your own with the colours and other bits and pieces in the shipped themes, you can fairly easily create one of your own. Have a look at the README file in the roxen/server-version/config_interface/themes directory of your installation (where version is the version of your server, e g 2.2.198), copy one of the present theme directories and put it in roxen/local/config_interface/themes/ instead, under a new name, and you're off! Putting it in the local/ hierarchy instead of the server-* tree makes sure your theme is still there when you upgrade to a newer Roxen version. Make sure you change the contents of the name file too to whatever you want to show up in the Theme selector.

Font Sizes

You may freely modify the font sizes used in the docs and for the various form elements of the administration interface with this setting. This setting only affects the contents used in the admin interface sent as text; buttons and other images rendered with Roxen's graphics engine remain their original sizes.

Note!

If you are using an old Netscape browser and consider mostly all fonts to be way too big, it may be an even wiser choice to tell it not to render fonts that big for unicode characters; see Page charset below.

Compact site list

If your server hosts hundreds or perhaps thousands of sites, the overview of all sites below the Sites tab tends to grow into a frightfully long and rather intimidating piece. The compact mode version replaces the site name buttons with a single select box for choosing your site.

Page charset

Being an internationalized product, Roxen supports sending its pages in foreign character encodings of various kinds - for Americans or western Europeans, ISO-8859-1 may be quite enough all or close to all of the time, but for encoding Japanese, Chinese, Vietnamese or Korean characters (to name but some) this is not enough, and thus the administration interface by default sends all its pages encoded in UTF 8, a common Unicode encoding. You may choose to use another encoding that suits you better here, and the characters that can not be expressed within that will be coded as HTML entities instead.

Note!

Some Netscape users may find that all fonts in the administration interface tend to get overly large when the utf-8 setting is used. This is most likely because the font settings under Edit -> Preferences -> Fonts has different settings for different encodings, and the default settings for the encoding Unicode is larger than the rest. Try modifying to your taste, if this is the case.

Interface language

If you would prefer Roxen administration in some other language than American English (or something close to it at least), this is where you can flip the switch to something you prefer over it.

You might notice that many things will still show up in English even after changing languages. If you want to help translating the interface to your language, we will happily receive your contributions and commit your work to future versions of Roxen. See the README file in your roxen/server-version/translations/ directory (where version is the version of your server, e.g. 2.2.198) for more details.

Font

Changing the administration interface font will not automatically switch fonts for all text of the admin interface, however all the admin interface's buttons and images rendered using Roxen's graphics support will be.

Note!

If you choose a font that lacks characters that are used by the translation to the interface language of your choosing or that you are using in site names for that matter, you may notice empty buttons and icons popping up all over the place. To fix this, either revert to the "roxen builtin" font, or another Unicode font.

Add/Delete module page type

With this setting, you may change the layout of the dialogue used for adding modules to a site; everything from the normal categorized, two-level folded list with descriptions to a very simplistic view where you can add lots of modules in one go. Try out the different options to find out what you like best.

2.2.2 Users

This tab contains the accounts for all users who have access to the administration interface, as well as their security clearance - what actions they may and may not perform in the admin interface of your server. Using the first two buttons, you create or delete users (both prompt you for a user name). A new user has no admin rights at all, so you will have to add them to your liking too. After changing the settings below to your liking, be sure to finalize them by clicking a Save button. The data fields for each user (identified by their login name heading) and their purpose:

Real name

The Real name you provide in the first field will be shown at the top right when the user is logged in, and also shows up in the debug log when he or she logs in, for instance "15:39:48 : Buttercup Utonium logged on as buttercup from buttercup.roxen.com (194.52.182.224).".

Password / Again

To change to a new password, write it in these fields. You have to write it twice, to be sure you spell it exactly the way you intended. Case matters. You will be prompted to rewrite the password if the two did not match.

Crypted

If you have a password crypted with the standard unix crypt() method, (for instance as fetched from a non-md5 /etc/passwd or /etc/shadow file) you want to set for the user, paste it in here instead of a plain-text password. This way, you as an administrator need not know what password the user really use - which is a good security practice.

Apart from the text fields, there is also an assortment of buttons. The Save button saves all changes made to the users in the list. Actually, all of the buttons in the matrix to the right are implicit save buttons too, so a change of user rights does not have to be confirmed with another

click on Save. The more prominent buttons on the right determine how much fun the user can have, or how much damage she can do, depending on your point of view.

All Permissions	Site: Administration Interface
View Settings	Update Client
Edit Global Variables	Tasks
Restart	Shutdown
Create Sites	Add Modules
Site: mysite	

The matrix of administration interface user permissions

All Permissions

Assuming the user is supposed to have all permissions available, use this shorthand instead of manually checking all of the others.

View Settings

To be allowed to access anything at all in the admin interface, apart from user settings, the Home and Docs tabs, a user needs View Settings rights.

Update Client

This flag grants the user rights to use the Update Client.

Site: [your site name here]

If you check one of the Site buttons, the user will have the rights to administer that site's configuration.

Hilfe

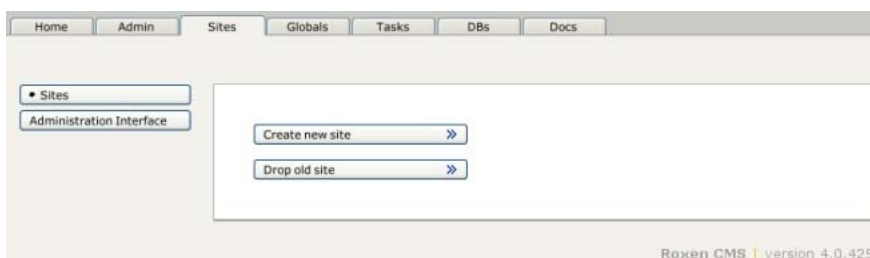
Checking Hilfe permissions gives this user the right to login to a hilfe port, thus being able to execute any pike code as the user running roxen via a telnet session. This can be a very handy feature for a developer, and an even greater security hole if used by a malicious user.

Note!

If a Hilfe port is setup not to require authentication at all, the configuration of this flag does not matter - everybody that connect will be allowed to use the hilfe interface.

2.3 Sites Tab

The Sites tab contains all web and ftp sites handled by the Roxen server. Most of the configurations are done here.



Sites tab

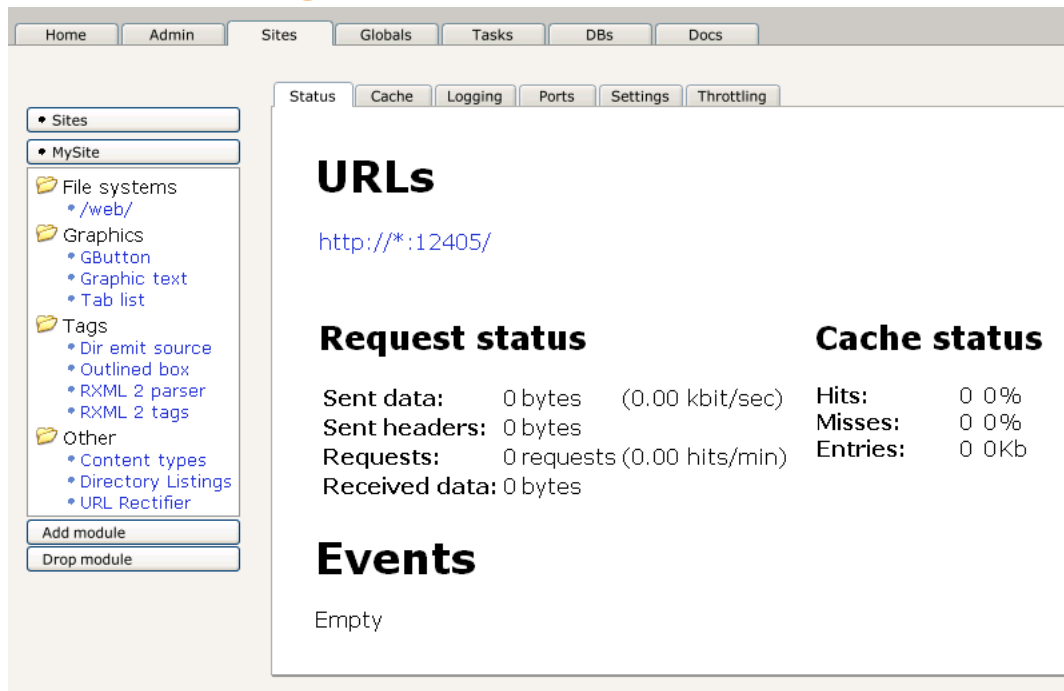
The Sites tab begins with a listing of the web and ftp sites to the left. It is possible to call up the site page for a site by clicking on the corresponding button.

Note!

The Administration Interface site is special since it contains the actual administration interface.

It is also possible to create a new site or delete a site with the Create new site and Drop old site buttons. See the Sites chapter for more information.

2.3.1 Site Page



Site page

The Site page contains information about a specific web- or ftp site. It lists all modules that make up the site as well as event log information related to the site.

It is possible to add or delete modules with the Add module and Drop module buttons, as well as focus on a module by clicking on the module's name. It is also possible to change the site settings from here, by choosing one of the tabs.

The Site settings page contains the site settings that are described in the Sites chapter. The most important settings are the URLs and the primary server URL variable, which control under what name and protocol it is possible to access the site. The actual port configuration is done under the Ports tab. The URL and port system, as well as the settings, are described in detail in the Ports chapter.

2.3.2 Module Page

The screenshot shows a web administration interface with a sidebar on the left and a main content area on the right. The sidebar contains a tree view with categories like 'Sites', 'File systems', 'Graphics', 'Tags', and 'Other'. The 'Graphics' category is expanded, showing 'GButton' as a sub-item. The main content area has tabs for 'Status', 'Settings', 'Security', and 'Documentation'. The 'Status' tab is active, displaying information about the 'Graphics: GButton' module. The information includes a description, cache statistics, tasks, developer information, and an inheritance tree.

Graphics: GButton

Provides the `<gbutton>` tag that is used to draw graphical buttons.

Images in cache: 116 images
Cache size: 194,6 kb

Tasks

[Reload](#) [Clear cache](#) [Drop Module](#)

Developer information

Identifier: gbutton
Thread safe: Yes ([more info](#))
Number of accesses: 0
Type: MODULE_TAG (`<gbutton/></>` and `<gbutton-url/></>`)
Loaded from: modules/graphics/gbutton.pike
CVS Version: 1.111 2004-06-30 16:59:03 mast

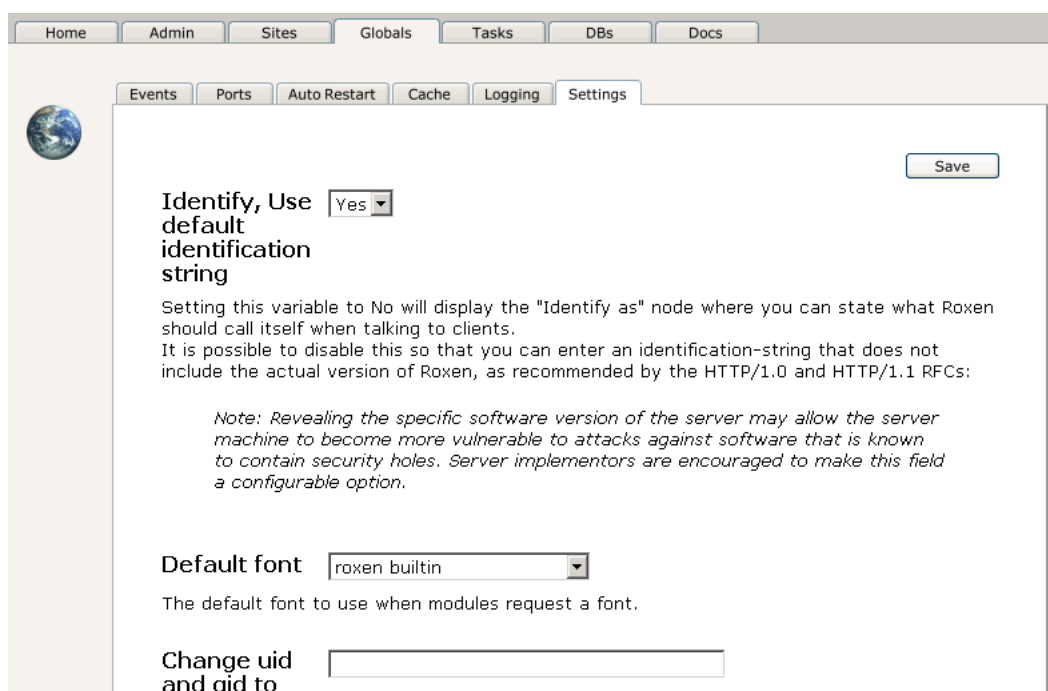
Inherit tree

modules/graphics/gbutton.pike
base_server/module.pike (version 1.216) (*No precompiled file available*)
base_server/basic_defvar.pike (version 1.30)

Module page

The module page contains information about the module itself, actions that can be done on that module (Reload, Drop module as well as module specific actions) and settings. It is described in more detail in the Modules chapter.

2.4 Globals Tab



The screenshot shows the 'Globals' tab in the Roxen Administration Interface. The top navigation bar includes 'Home', 'Admin', 'Sites', 'Globals', 'Tasks', 'DBs', and 'Docs'. Below this, a sub-navigation bar contains 'Events', 'Ports', 'Auto Restart', 'Cache', 'Logging', and 'Settings'. The 'Settings' sub-tab is active, displaying a 'Save' button in the top right corner. The main content area features a section titled 'Identify, Use default identification string' with a 'Yes' dropdown menu. Below this, explanatory text states that setting this to 'No' will display an 'Identify as' node, and it is possible to disable this to enter an identification-string that does not include the actual version of Roxen, as recommended by the HTTP/1.0 and HTTP/1.1 RFCs. A note follows, warning that revealing the specific software version may allow the server machine to become more vulnerable to attacks against software that is known to contain security holes, and encourages server implementors to make this field a configurable option. Below the note, there is a 'Default font' dropdown menu set to 'roxen builtin', with a description: 'The default font to use when modules request a font.' At the bottom, there is a 'Change uid and gid to' text label followed by an empty input field.

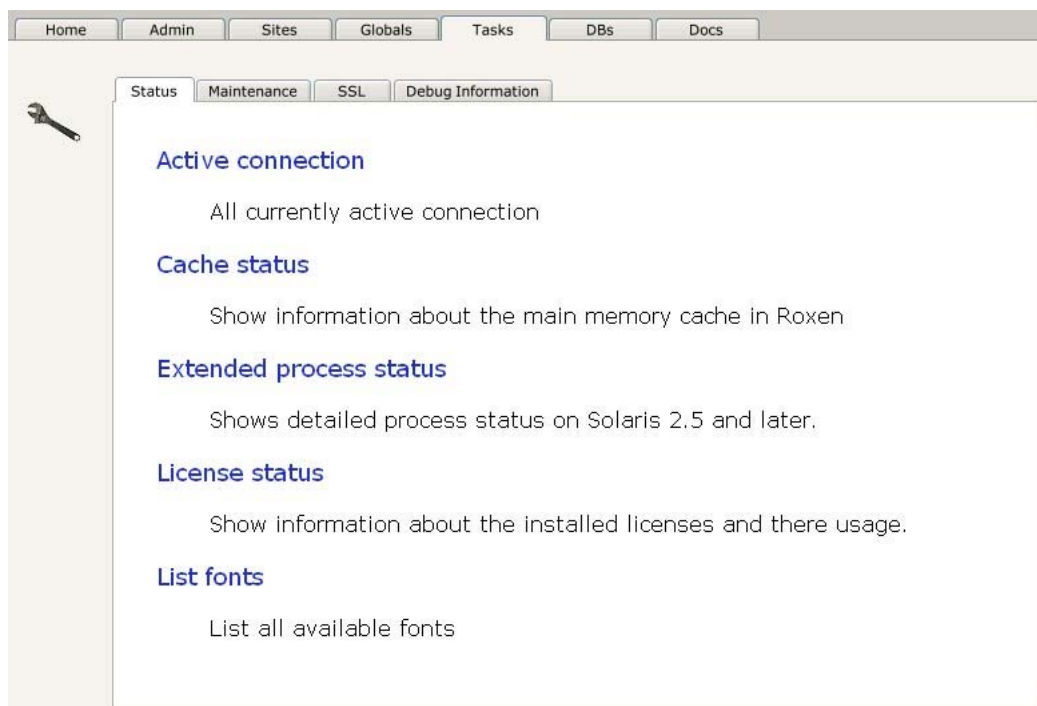
Globals tab

The Globals tab contains settings that affect the whole Roxen server. They are described in the Globals chapter.

Note!

Not all global settings are handled by the Globals tab, some special options are only available as options to the start script.

2.5 Tasks Tab



The screenshot shows the 'Tasks' tab in the Roxen Administration Interface. The top navigation bar is the same as in the previous screenshot. The sub-navigation bar includes 'Status', 'Maintenance', 'SSL', and 'Debug Information'. The 'Status' sub-tab is active, displaying a list of status-related tasks. Each task is preceded by a blue link: 'Active connection', 'Cache status', 'Extended process status', 'License status', and 'List fonts'. Each link is followed by a brief description of what the task shows or lists. For example, 'Active connection' shows all currently active connections, 'Cache status' shows information about the main memory cache, 'Extended process status' shows detailed process status on Solaris 2.5 and later, 'License status' shows information about installed licenses and their usage, and 'List fonts' lists all available fonts.

Tasks tab

The Tasks tab contains support wizards that are not related to configuring the server but some other task, such as restarting the server, flushing memory caches, creating SSL certificates or getting debug information.

2.6 DBs Tab

The DBs tab, unsurprisingly, empowers you to administer any databases you want to use from the sites handled by roxen server; from creating and grouping up to moving, backing up, browsing and tweaking or dropping databases at your leisure. You will also find some info on the backend MySQL database used internally by Roxen.

Below the main subtab, Databases, you find a list of the registered databases and what sites have what access rights to their contents; N for no rights, R for read and W for both read and write rights. At the far right is indicated whether the database resides in the internal database - the MySQL database shipped with Roxen, or if it's a remote, external database, and finally how big its contents is, altogether.

2.6.1 Database Groups

The databases are grouped together in database groups by category; there's the ones used by Roxen itself for the documentation tab and the local storage (holding all sorts of caches, for example), and then there may be any number of other groups created by modules or yourself as an administrator for your own purposes.

You create a database group of your own by clicking the Create new group button. If you intend to have several databases created or imported right away from a particular MySQL server, you may write a database URL to that server, leaving out the (already filled-in) mysql:// part, for example root@dbhost. This will prompt you with a list of all the databases available at dbhost to import into the group right away, relieving you of some boring work.

The group name won't show up in the main view until you have put some databases in it, either by importing this way, or by creating a new database, assigning it to the group. Be sure to come up with intuitive group names do write good comments of the purpose of your database groups, as a reminder to yourself and the co-administrators of your server, if any; they will most likely be happy you did. :) You may review this comment at any time by clicking on the database group name in the Databases view.

2.6.2 Databases

Each database name on the left on the Databases tab is a clickable that link zooms in on the database in question, showing its database URL, the tables it holds, how many rows they have and the total size of the table's data - you may resort the list by clicking on either column. Clicking on a table reveals a brief list of its fields.

Below the list is a text field where you may write SQL queries of your own, to browse or alter its contents, or perform any other administrative actions your database engine might support via its SQL interface. The query is run right away when you press the Run query button, and its results printed below the text field on the result page. When zooming down on a table you haven't already performed a query on, the query field suggests listing all columns declared as text, integers and similar readable data. Should you select blobs that turn out to be images, Roxen graciously presents them to your browser as image links instead of huge blobs of binary data - try looking at the image cache for GButton in the local database, for instance.

2.6.3 Creating New Databases

At the bottom of the Databases page, you can create or add new databases to the list with the Create new database button. The name you choose for the database will be the name you use as the db attribute of queries using the <emit sql> tag later on, so it's a good idea to use a name that is easy to type and identify the database for the RXML author.

An internal database will be both created (in the bundled internal MySQL database) and added to the view, whereas an external database will just be added - if it didn't exist, an error is reported. When creating an external database, you must fill in a database URL to the database server (e.g. `mysql://root@dbhost/`), or to a particular database itself (such as `mysql://root@dbhost/customers`). The database will be listed below the category of your choice, and when you create internal databases, the category can also decide in what database server the database will be created. Again, you are encouraged to add a helpful comment about the database; it will be shown when browsing the database later on, so here is where you write the reminders you will later wish you had given the effort to write, trying to get reacquainted with the dusty old database from last year.

2.6.4 Database Buttons

When focusing a database, a new array of action buttons is made available:

Make a backup

For internal databases, you may choose to make a backup of its data and structure to be restored manually at a later date in case of the unexpected. All database backups show up under the Database Backups subtab to be restored or deleted at your leisure. Roxen will remember from what database it was backed up, but you may also read them back into some other database if you wish.

Copy or move this database

A copy of the database contents being made somewhere else, either in an internal or external database. If you choose to copy the database, the source database will not be dropped from the database server, but if you move it and it, and the source is an internal database, it will be purged.

Delete this database

The database is removed from your database view. When deleting an external database, no data is actually touched at the remote database server - this is only done with internal databases.

Delete all tables

Regardless of whether the database is external or internal, this drops all of its tables in one go. As a safety measure, you will have to confirm this dangerous action before it is actually performed.

Change group for this database

If you should decide you wanted to regroup a database, this action comes in handy.

2.6.5 Database Backups

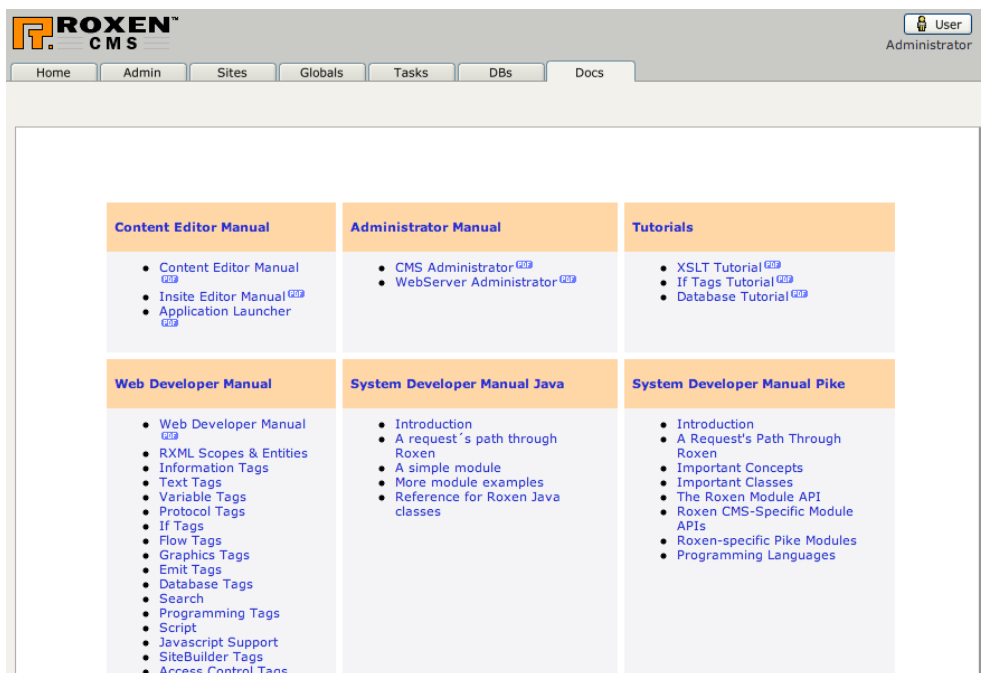
All databases backed-up using the Make a backup button above end up below this subtab, and upon restoring one, you are prompted of what tables you want and what database to read them back to. If you want to read them back to a new database, be sure to create one first. The default choice is the source database it was backed up from.

In the list, organised by database name, you will see the paths to all database backups, when they were made, their name and buttons to restore or delete them. Deleting a backup is performed with no questions asked, so be sure to know what you are up to.

2.6.6 MySQL Status

The last subtab, finally, lists current status info and details on what the bundled MySQL server is up to, version information, active connections and various other useful debugging data.

2.7 Docs Tab



Docs tab

The Docs tab contains the online documentation. The latest version of the documentation for Roxen WebServer, the free part of Roxen CMS, can be found at roxen.com.

2.8 Server Maintenance

This chapter covers using command line options when starting Roxen CMS and custom options when compiling Roxen CMS from source.

2.8.1 Using the configure command

This command is used exclusively when installing Roxen CMS from source. Check the Software Installation for further instructions on installation from source.

2.8.2 Using command line options

The section covers passing various options when starting Roxen CMS. This is mainly used for debugging purposes.

Unix/Linux example:

```
root@www:~/usr/local/roxen$ ./start --once --remove-dumped
: Maximum number of open FDs: 1024
: Server start command:
:   /usr/local/roxen/server-4.0.425/bin/roxen
:   -DMODULE_DEBUG
```

```

:      -DRAM_CACHE
:      -DENABLE_THREADS
:      -M/usr/local/roxen/server-4.0.425/etc/modules
:      -M/usr/local/roxen/local/pike_modules
:      -I/usr/local/roxen/server-4.0.425/etc/include
:      -I/usr/local/roxen/server-4.0.425/base_server
:      -P/usr/local/roxen/server-4.0.425
:      -w
:      base_server/roxenloader.pike
:      --once=1
:      --pid-file=../configurations/_roxen_pid
11:30:32 : -----
0m 0.0s : Pike version:      Pike v7.4 release 340
: Product version:   Roxen CMS 4.0.425-release5
: Operating system:  Linux 2.6.12-10-amd64-k8 (x86_64)
: Starting MySQL ... Done 4.0.18-standard [148.1ms]

```

Microsoft Windows:

Passing command line options can be done in two ways. Either by setting additional parameters in the \roxen\local\environment.ini file or by setting the options when starting the Roxen CMS from the command line.

2.8.3 Options for ./start

--version

Display version information.

--help

Show help text.

--remove-dumped

Depending on configuration, Roxen may attempt to dump compiled pseudocode binaries of Pike files to avoid having to recompile them next time Roxen restarts. This option removes such files, making sure Roxen recompiles all files from the source code. This should not be necessary in normal operation, but may be useful when developing or debugging Roxen modules.

--verbose (short form: -v)

Give more diagnostics and progress information than usual.

--quiet (short form: -q)

Give less diagnostics and progress information than usual.

--offline

Disables DNS lookups and some other code that requires a net connection (most content boxes in the admin interface, for instance). In --offline mode, you can only bind ports to IP addresses (e.g. http://127.0.0.1/) or "*", unless your system has non-DNS means of resolving hostnames to IP addresses. On a unix system, this could be achieved by adding required host names to /etc/hosts.

--log-dir=DIRECTORY

Put log files in given directory. The default is '../logs'.

--config-dir=DIRECTORY

Look for configuration files in the given directory. The default is '../configuration'.

--with-threads or --without-threads

Whether to use threads (i.e. run several request handler threads in parallel) or not. The default depends on operating system.

--once

Only run Roxen once, and send the debug log to standard output. This is primarily useful for semi-interactive debugging, and means Roxen will not restart automatically when the process ends (even if it ends because the 'Restart' action was requested from the configuration interface).

This argument also implies `--module-debug`.

--with-keep-alive or --without-keep-alive

Whether to enable keep-alive request handling in the HTTP protocol. The current default is to enable keep-alive.

--with-debug or --without-debug

Whether to enable extra debug information or not.

Some further arguments can be given, which will not directly affect Roxen but rather be passed to the Pike binary that will run Roxen. This is mainly useful for enabling various forms of extra debugging.

-Dsymbol

Define the specified symbol for the Pike preprocessor, as if it had been defined as `'#define symbol'` in all Pike source files.

-Mpath

Set the Pike module path, i.e. the directory where Pike will look for Pike modules. (Do not confuse this with the path where Roxen looks for Roxen modules written in Pike. That path is configured through the configuration interface.)

-lpath

Set the Pike include path, i.e. the directory where Pike will look for files when doing `#include`.

-Ppath

Set the path to the Pike binary.

More options are available, but they are mainly useful for developers and for debugging purposes. Do `'./start -help'` for a complete (but more terse) list of options for the start script.

2.9 Sites



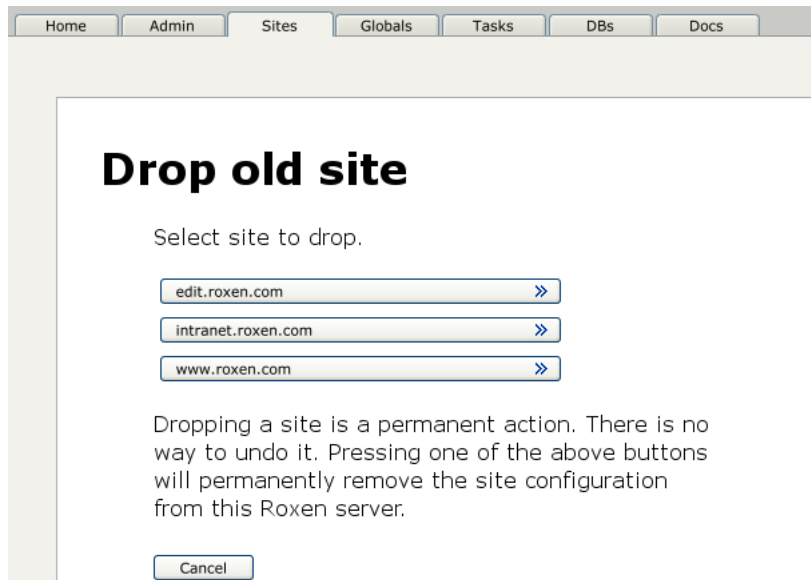
Sites tab

The Sites tab contains all your web- and ftp sites. It is here that sites are created, deleted and configured. For a web- or ftp site to be accessible from the outside world it needs to bind to a port. How the ports work is explained in the Ports chapter.

2.9.1 Create a New Site

How to create a new site is described in the Installation chapter.

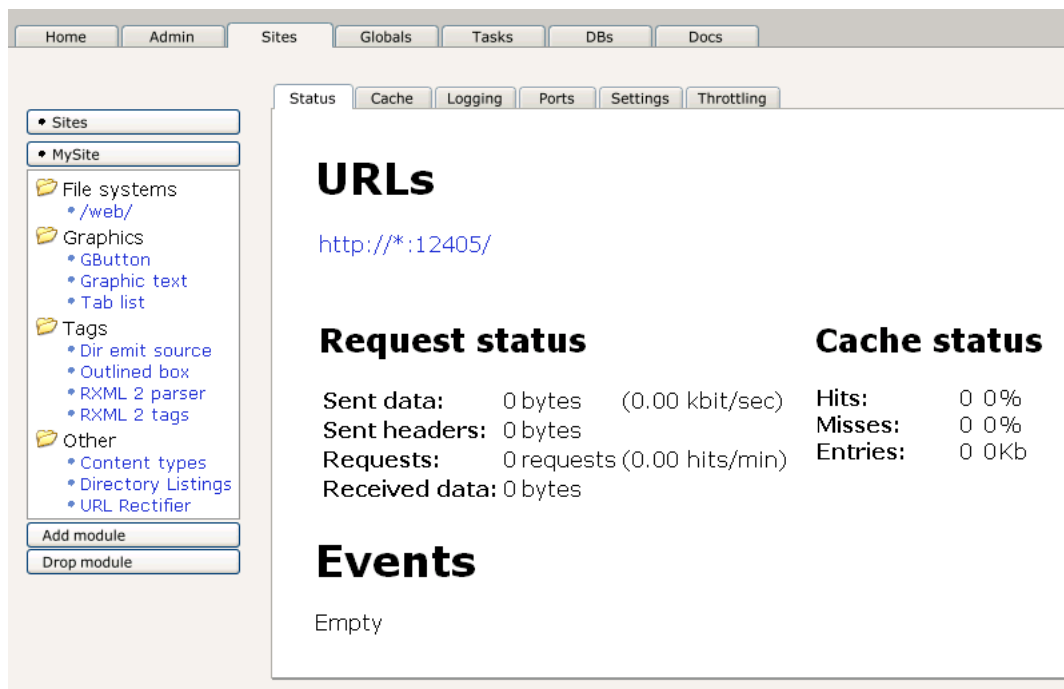
2.9.2 Delete a site



Drop site

To delete a site press the Drop old site button and choose which site to delete. In the case of the site being a CMS site, you will also be given the options of removing the corresponding databases or not. By default the databases are deleted.

2.10 Site Information



Site page

The site page contains information about the site as well as tabs for site settings and links to all modules. Exactly what information is shown is controlled by the settings under the Admin / Your settings tab. From the site page it is possible to configure the site settings, add or delete modules with the Add module and Drop module buttons as well as focusing on a module by following the module's link. The module related buttons are described in detail in the Modules chapter.

2.10.1 URLs

URLs lists the URLs that this site is available as. The first link links to the site itself, the second link (handled by) links to the port that handles this URL. The port is under the Ports tab and is described in detail in the Ports chapter.

2.10.2 Request Status

The request status gives the number of requests and amount of data sent and received by this site since the server was restarted. Since it only counts since the server was restarted it is not always relevant, the full information is available in the access log file, which is described in detail in the Logs chapter.

2.10.3 Cache Status

Cache status gives information about how well the data cache has performed for this site. The data cache is used to get better performance for frequently visited pages of files. The Data cache tab controls the cache. It is described in detail further down.

Only sites with high traffic need the data cache, for normal sites the performance is enough without it.

2.10.4 Events

Events lists the part of the event log that relates to this site. The full event log is available under the Events tab. All logs are described in detail in the Logs chapter.

2.11 Site Settings

The screenshot shows the 'Site Settings' page in the Administration Interface. The top navigation bar includes 'Home', 'Admin', 'Sites', 'Globals', 'Tasks', 'DBs', and 'Docs'. The 'Sites' tab is active, and the 'Settings' sub-tab is selected. On the left, a sidebar lists various modules under 'roxen standard', including 'Access Control', 'Authentication', 'CMS', and 'CMS Editor Component'. The main content area contains the following settings:

- Default site:** A dropdown menu currently set to 'No'. A 'Save' button is located to the right.
- Site comment:** A large text area for entering a comment. Below it, a note states: 'This text will be visible in the administration interface, it can be quite useful to use as a memory helper.'
- Site name:** A text input field. Below it, a note states: 'This is the name that will be used in the administration interface. If this is left empty, the actual name of the site will be used.'

Site settings

The site settings are divided into six tabs: Status, Cache, Logging, Ports, Settings and Throttling. The most important settings are under the Ports and Settings tabs. Throttling controls the bandwidth throttling system that makes it possible to limit how much bandwidth is used by the site.

Cache controls the data cache that gives better performance to high traffic sites. Logging controls what is written to the log files and is described in detail in the Logs chapter.

2.11.1 Ports

The Primary Server URL is the URL that is used when the site is referring to itself, for example when performing redirects. Modules frequently read this setting when creating URLs to the site. If the site can be reached by several URLs this should be the preferred one.

The important site settings control how the site is accessible from the world. The URLs variable controls the URLs this site binds to. How a URL is translated into a port is described in the Ports chapter.

The remainder of the Ports tab deals with specific settings per ip number.

2.11.2 Settings

Default site controls which site will be chosen if there are several to choose from. This is described in more detail in the Ports chapter.

Domain is the domain name of the site, used by modules in the site when they need to refer to a domain. It is, for example, used to create email addresses by some RXML tags.

No such file message contains the error message sent if a page could not be found, and makes the Page not found message configurable. It should be changed to something more appropriate for the site itself, rather than being a standard message. For example something linking to a commonly viewed page on the site.



The standard 404 message

2.11.3 Throttling

The bandwidth throttling settings make it possible to limit the bandwidth given to each site and to each request. A number of modules can be used to tune the actual bandwidth throttling scheme. See the Modules chapter for more information about them. The bandwidth throttling system must be enabled before these modules can do anything.

2.11.4 Cache

The data cache is used to cache static pages that get accessed a lot. The RXML parser figures out whether a page is static or not by letting each tag tell whether it can be cached or not.

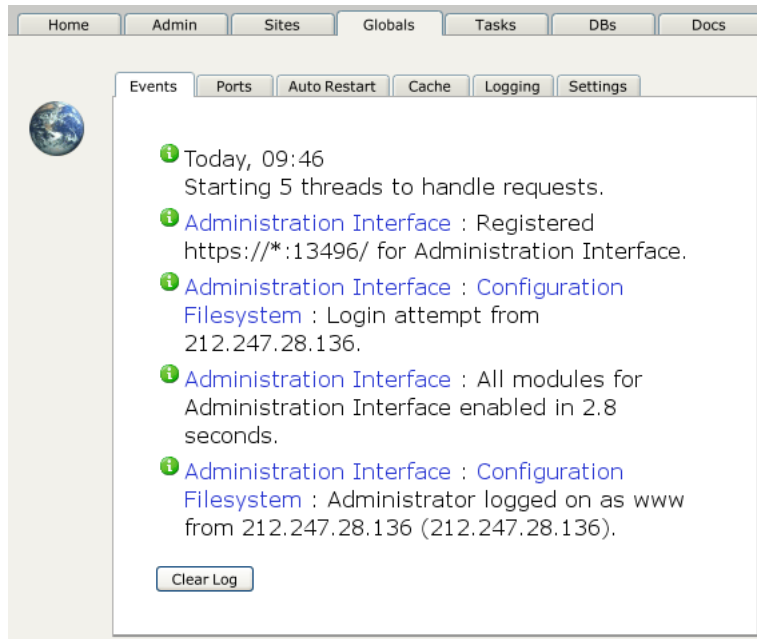
The data cache is used to speed up delivery of pages that are accessed frequently. The configuration options are Cache size which is the maximum size of the data cache and Max file size which is the maximum size of a file in the cache.

Only sites with high traffic need the data cache.

The Cache status header on the site page shows how well the data cache is performing as well as how much memory it is using.

There are other caches in Roxen as well, though the data cache is the only one that stores full files. The Status / Cache status task shows information about all the caches.

2.12 Logs



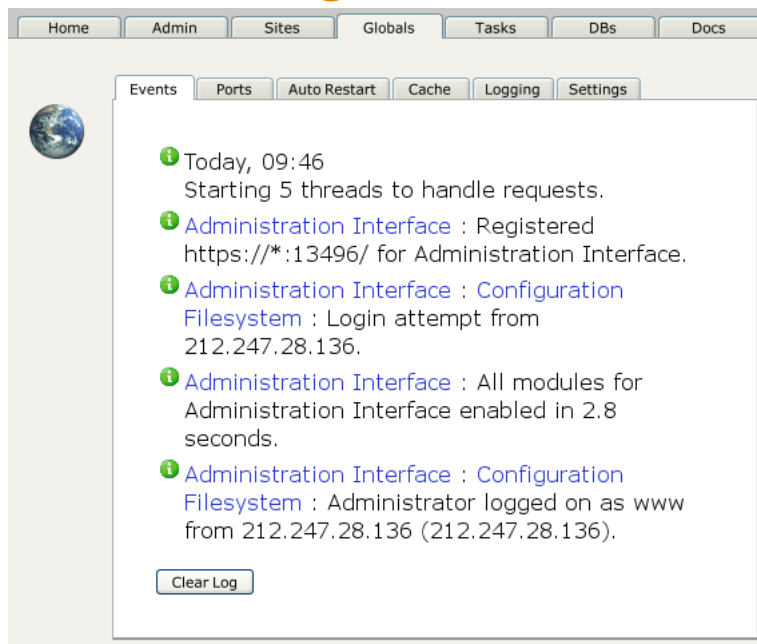
Event log

Roxen usually logs errors as well as all accesses to its sites. The errors are logged in the event log as well as in the debug log. Accesses are logged to each site's access logs. It is also possible for modules to perform logging.

LogView can be used to analyze the access log, or in place of the access log altogether. LogView is described in the LogView chapter.

The event log is a web based error log available in the administration interface. It is available both for the whole server as well as for each site and module.

2.12.1 Event Log

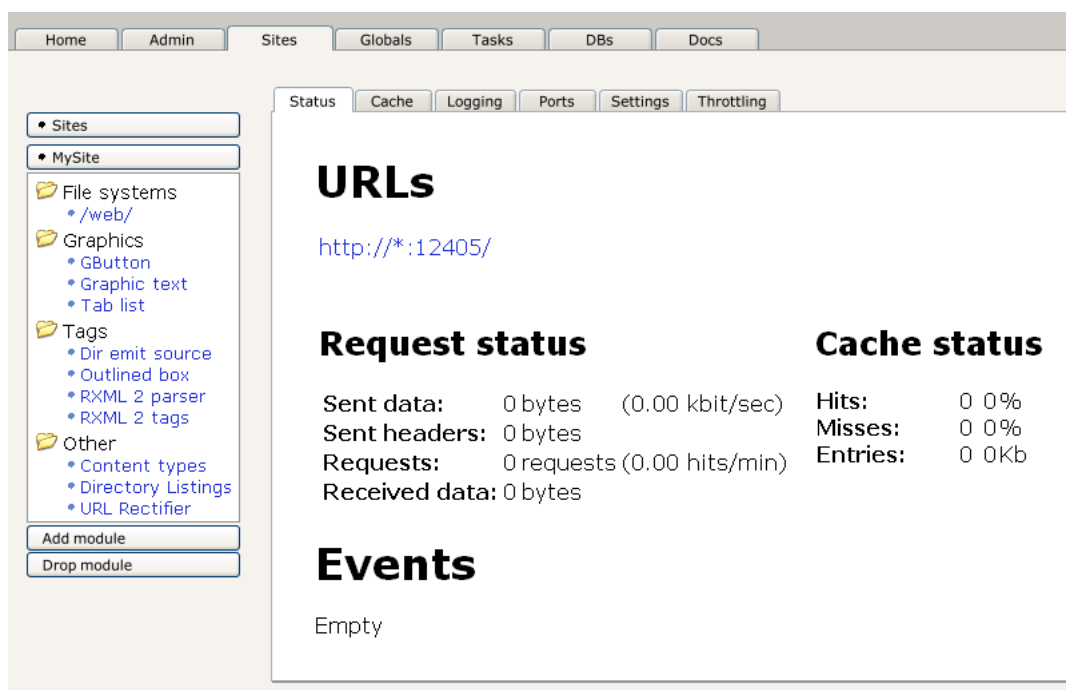


Full event log

The event log is a web based error log. It uses icons to signal if it is an error, a warning or only an information message. Each log entry has links to the site and/or module the message refers to. Everything in an attempt to create a log that is as easy to read as possible.

The event log is the preferred way for a Roxen administrator to find out if her server is working properly. The full event log is available on the Events tab. Relevant portions of the event log is also available on each site page and each module page.

The event log is cleared by pressing the Clear log button. It is a good idea to clear the log after having made sure it does not contain any serious errors. That way the entire log will be new next time you look at it.



Event log on site page

The screenshot shows the Roxen CMS Administration Interface. At the top, there are tabs for Home, Admin, Sites, Globals, Tasks, DBs, and Docs. Below these, there are sub-tabs for Status, Settings, Security, and Documentation. On the left side, there is a sidebar with a tree view showing the site structure: Sites, MySite, File systems (with a link to /web/), Graphics (with sub-items GButton, Graphic text, and Tab list), Tags (with sub-items Dir emit source, Outlined box, RXML 2 parser, and RXML 2 tags), and Other (with sub-items Content types, Directory Listings, and URL Rectifier). Below the sidebar are buttons for 'Add module' and 'Drop module'. The main content area is titled 'Graphics: GButton' and contains the following information:

- Provides the <gbutton> tag that is used to draw graphical buttons.**
- Images in cache:** 116 images
- Cache size:** 194.6 kb
- Tasks:** Reload, Clear cache, Drop Module
- Developer information:**
 - Identifier:** gbutton
 - Thread safe:** Yes ([more info](#))
 - Number of accesses:** 0
 - Type:** MODULE_TAG (<gbutton/></> and <gbutton-url/></>)
 - Loaded from:** modules/graphics/gbutton.pike
 - CVS Version:** 1.111 2004-06-30 16:59:03 mast
- Inherit tree:**
 - modules/graphics/gbutton.pike
 - base_server/module.pike (version 1.216) (No precompiled file available)
 - base_server/basic_defvar.pike (version 1.30)

Event log in module page

2.12.2 Debug Log

```
14:30:35 : -----
0m 0.0s : Pike version:      Pike v7.4 release 340
        : Product version:   Roxen CMS 4.0.425-release5
        : Operating system: Linux 2.6.12-10-amd64-k8 (x86_64)
        : Starting MySQL ... Done 4.0.18-standard [672.3ms]
        :
14:30:36 : mysql: Warning: One can only use the --user switch if running as
root
0m 0.7s : mysql: 060127 14:30:35 InnoDB: Started
        : mysql: bin/roxen_mysql: ready for connections.
        : mysql: Version: '4.0.18-standard' socket:
'/usr/local/roxen/configurations/_mysql/socket' port: 0
```

Beginning of debug log

The debug log is available as `roxen/logs/debug/default.1`. The debug log contains essentially the same information as the event log, but in a text only format. It is sent either to a file or to syslog on a unix system. This is controlled by the Logging / Debug log method variable under the Globals tab.

The debug log is important when the server starts, since it writes messages before the administration interface becomes available. If the server did not restart properly the debug log will show why.

The debug log also contains the process id of the Roxen process and the start script process. This information is important if you want to restart the server without using the admin interface, for instance by using the kill command instead.

2.12.3 Access Log

Roxen usually logs all accesses to a site to the access log, which is located at roxen/logs/server name/Log.


```
dhcp4.idonex.se - - [20/Sep/2000:15:39:52 +0200] "GET
/_internal/gbutton!0/ea$c6 HTTP/1.1" 200 1464
dhcp4.idonex.se - - [20/Sep/2000:15:39:52 +0200] "GET
/_internal/gbutton!0/29$e2 HTTP/1.1" 200 1324
dhcp4.idonex.se - - [20/Sep/2000:15:39:52 +0200] "GET
/_internal/gbutton!0/29$eb HTTP/1.1" 200 1522
dhcp4.idonex.se - - [20/Sep/2000:15:41:10 +0200] "GET /standard/actions/
HTTP/1.1" 200 4525
dhcp4.idonex.se - - [20/Sep/2000:15:41:11 +0200] "GET
/_internal/gbutton!0/59$4d HTTP/1.1" 200 1333
dhcp4.idonex.se - - [20/Sep/2000:15:41:11 +0200] "GET
/_internal/gbutton!0/93$9b HTTP/1.1" 200 1606
dhcp4.idonex.se - - [20/Sep/2000:15:41:11 +0200] "GET
/_internal/gbutton!0/93$cd HTTP/1.1" 200 1247
dhcp4.idonex.se - - [20/Sep/2000:15:41:11 +0200] "GET
/_internal/gbutton!0/ea$10 HTTP/1.1" 200 1852
dhcp4.idonex.se - - [20/Sep/2000:15:41:13 +0200] "GET
/standard/actions/?class=debug_info HTTP/1.1" 200 4885
dhcp4.idonex.se - - [20/Sep/2000:15:41:13 +0200] "GET
/_internal/gbutton!0/a4$5b HTTP/1.1" 200 1386
```

Part of an access log

The access log is controlled by settings under both the Globals and the Sites tab. Under the Globals / Logging tab are log settings that affect all sites, while the Logging tab under each site contains settings that only affect that site.

Home
Admin
Sites
Globals
Tasks
DBs
Docs

Events
Ports
Auto Restart
Cache
Logging
Settings



Save

Log entire file length in restored connections
No

If this toggle is enabled log entries for restored connections will log the amount of sent data plus the restoration location. Ie if a user has downloaded 100 bytes of a file already, and makes a Range request fetching the remaining 900 bytes, the log entry will log it as if the entire 1000 bytes were downloaded.

This is useful if you want to know if downloads were successful (the user has the complete file downloaded). The drawback is that bandwidth statistics on the log file will be incorrect. The statistics in Roxen will still be correct.

Audit trail
No

If Audit trail is set to Yes, all changes of uid will be logged in the Event log.

Debug log method
file

What method to use for the debug log, default is file, but syslog is also available. When using file, the output is really sent to stdout and stderr, but this is handled by the start script.

Log directory prefix
../logs/

This is the default file path that will be prepended to the log file path in all the default modules and the site.

Save

Globals / Logging

The Logging tab under the Globals tab include settings for the directory where all log files are stored.

Home Admin Sites Globals Tasks DBs Docs

• Sites

• www.roxen.com

Access Control

- Basic HTTP and FTP authentication
- Cookie HTTP authentication
- Internal password
- Main module

CMS

- Content Editor
- Insite Editor: www.roxen.com/ mounted on /
- Main Module
- Site News
- Tags 2.0
- Templates
- XSLTransform

CMS Editor Component

- File
- Footer
- Header
- Insert Component
- Link
- Menu
- News
- Table
- Text & Picture

Graphics

- Business graphics
- GButton
- Graphic text
- Image converter
- Tab list

Search

- CMS Advanced Interface
- Query Module

Status Cache Logging Ports Settings Throttling

Save

Enabled Yes

Log requests

Format

```
404: $host $referer - [$cern_date] "$method $resource
$protocol" 404 -
500: $host $referer ERROR [$cern_date] "$method $resource
$protocol" 500 -
*: $host - - [$cern_date] "$method $resource $protocol"
$response $length
```

What format to use for logging. The syntax is:

response-code or *: Log format for that response code

Log format is normal characters, or one or more of the variables below:

\n \t \r	-- As in C, newline, tab and linefeed
\$char(int)	-- Insert the (1 byte) character specified by the integer.
\$wchar(int)	-- Insert the (2 byte) word specified by the integer.
\$int(int)	-- Insert the (4 byte) word specified by the integer.
\$^	-- Supress newline at the end of the logentry
\$host	-- The remote host name, or ip number.
\$vhost	-- The Host request-header sent by the client, or - if none
\$ip_number	-- The remote ip number.
\$bin-ip_number	-- The remote host id as a binary integer number.
\$cern_date	-- Cern Common Log file format date.
\$bin-date	-- Time, but as an 32 bit integer in network byteorder
\$method	-- Request method
\$resource	-- Resource identifier
\$full_resource	-- Full requested resource, including any query fields
\$protocol	-- The protocol used (normally HTTP/1.0)
\$response	-- The response code sent
\$bin-response	-- The response code sent as a binary short number
\$length	-- The length of the data section of the reply
\$bin-length	-- Same, but as an 32 bit integer in network byteorder
\$request-time	-- The time the request took (seconds)
\$referer	-- The header 'referer' from the request, or '-'
\$user_agent	-- The header 'User-Agent' from the request, or '-'

Site / Logging

The Logging tab settings under Sites make it possible to configure exactly what information is put in the log. It is, for example, possible to log user names.

2.12.4 Accessed Counter

The screenshot displays the Roxen Administration Interface. At the top, there are tabs for Home, Admin, Sites, Globals, Tasks, DBs, and Docs. Below these, a sub-menu shows Status, Settings, Security, and Documentation. On the left, a sidebar lists various modules under categories like Sites, Access Control, CMS, CMS Editor, Graphics, Search, and Tags. The 'Accessed counter' module is selected under the 'Tags' category. The main content area shows the 'Tags: Accessed counter' configuration page. It includes a description of the module's purpose, a 'Tasks' section with 'Reload' and 'Drop Module' buttons, and a 'Developer information' section with details about the module's identifier, thread safety, number of accesses, type, and version. An 'Inherit tree' section shows the module's inheritance hierarchy.

Tags: Accessed counter

This module provides access counters, through the `<accessed>` tag and the `&page.accessed;` entity.

1 entries in the accessed database.

Tasks

[Reload](#) [Drop Module](#)

Developer information

Identifier: accessed
Thread safe: Yes ([more info](#))
Number of accesses: 0
Type: MODULE_TAG (`<accessed/>`)
MODULE_LOGGER
Loaded from: modules/tags/accessed.pike
CVS Version: 1.55 2005-04-21 11:22:28 erikd

Inherit tree

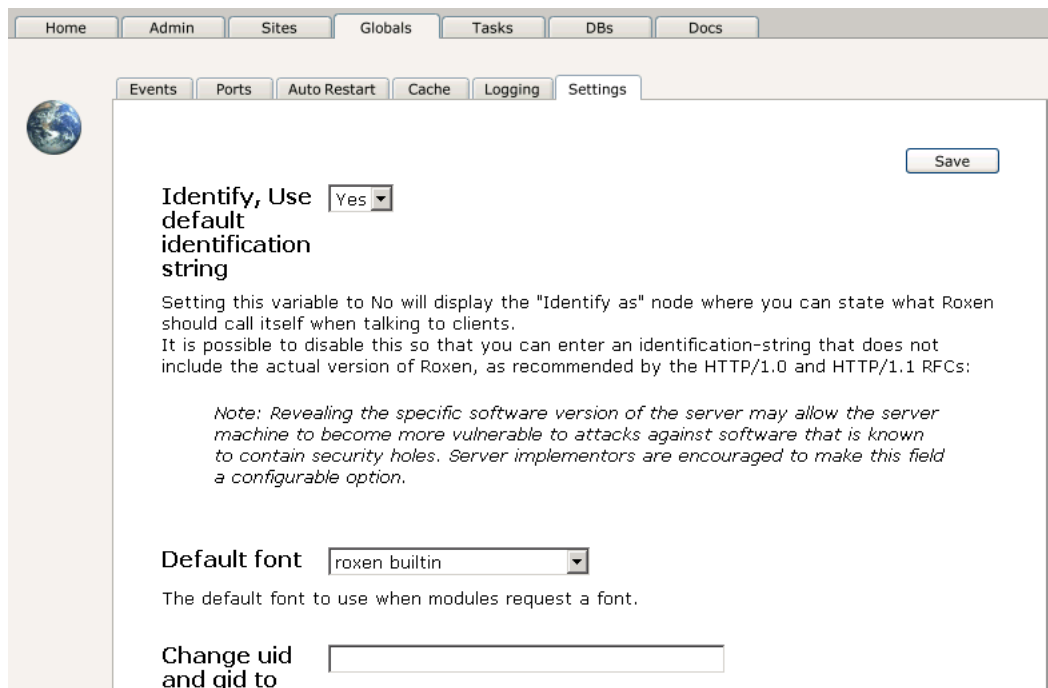
modules/tags/accessed.pike (version 1.55)
base_server/module.pike (version 1.222) (No precompiled file available)
base_server/basic_defvar.pike (version 1.30)

Accessed counter module

Roxen contains a module that makes it possible to have an accessed counter on any web page. The accessed counter module uses a database file in `roxen/logs/site name/Accessed` to store accesses in. By default it only stores accessed to pages that actually contain an `<accessed>` tag or `&page.accessed;` entity, but it can be configured to always count accessed to certain extensions.

Since the accessed counter functionality takes some CPU and memory it is not installed by default, but the administrator has to enable the Accessed counter module herself.

2.13 Globals



The screenshot shows the Roxen Administration Interface. At the top, there are tabs: Home, Admin, Sites, Globals, Tasks, DBs, and Docs. The 'Globals' tab is selected. Below this, there are subtabs: Events, Ports, Auto Restart, Cache, Logging, and Settings. The 'Settings' subtab is active. On the left side of the main content area, there is a small globe icon. The main content area contains the following settings:

- Identify, Use default identification string**: A dropdown menu set to 'Yes'. Below it, a paragraph explains that setting this to 'No' will display the 'Identify as' node, which allows stating what Roxen should call itself when talking to clients. It also notes that it's possible to disable this to enter an identification-string that does not include the actual version of Roxen, as recommended by the HTTP/1.0 and HTTP/1.1 RFCs. A note below states: *Note: Revealing the specific software version of the server may allow the server machine to become more vulnerable to attacks against software that is known to contain security holes. Server implementors are encouraged to make this field a configurable option.*
- Default font**: A dropdown menu set to 'roxen builtin'. Below it, a paragraph states: 'The default font to use when modules request a font.'
- Change uid and gid to**: An empty text input field.

A 'Save' button is located in the top right corner of the main content area.

Globals tab

The Globals tab contains settings that affect the whole web server and holds information that is not related to any single site. It consists of six subtabs; Events, Ports, Auto Restart, Cache, Logging and Settings.

Events

This tab holds all of the status and error messages reported by the modules running on the server.

Ports

Similar to Events, this tab is a collection of data that spans all sites on the server, notably the configuration and status of the ports they open.

Auto Restart

Schedules automatic restarts of the server at regular intervals.

Cache

Here you configure the memory cache and the proxy disc caches.

Logging

This is where you specify how, where and what gets logged for the debug and access logs. For more information about logging, refer to the Logs chapter.

Settings

The final global subtab contains settings for global server defaults, where the server should look for fonts and other server-global parameters.

2.14 Port System

2.14.1 Ports tab

Each site is available to the world on one or several URLs. The Port system, available under the Ports tab, handles the particulars of making the URL available. Each port handles one or several URLs. Ports are created automatically for the URLs. Sometimes no configuration is necessary at all, since Roxen tries to find out all information necessary to create a suitable port. However it is not always as simple. But to understand how a port is created we need to go describe how a resource is made available to computers on the Internet.

2.14.2 Protocols and the Internet

The protocol is the language that the computers talk to understand each other. Many different protocols are in use on the Internet, and at the bottom of practically all of them is the Internet Protocol (IP). On top of IP is TCP/IP and UDP/IP. UDP/IP is used for such services as the Network File System (NFS) and the Domain Name System (DNS), but most of the interesting protocols as far as Web servers are concerned are built on top of TCP/IP (or just 'TCP', for short).

TCP is oriented in terms of *connections* between *sockets*. A socket is identified by two things: its IP *address*, and its *port number*. A connection is identified by the two sockets it connects. In the normal case, one end of the connection is a server socket, and the other end a client socket. The port number at the *server* socket typically indicates which protocol (built on top of TCP/IP) that is in use on the connection. The port number at the *client* socket is used to tell different connections to the same remote service apart, and is of no particular significance to an application programmer or a web user.

The most central protocol for typical web traffic is the Hyper-Text Transfer Protocol (HTTP), closely followed by the encrypted version of the same thing (HTTPS). Roxen supports these two protocols directly for accessing web pages. A third option available is the traditional File Transfer Protocol (FTP). The standard port numbers used for these protocols are 80 for HTTP, 443 for HTTPS and 21 for FTP.

2.14.3 Numeric vs symbolic addresses

IP addresses are 32-bit numbers, often written in the form of four 8-bit components with periods between them, e.g. '127.0.0.1'. Since these names aren't always very easy to remember, a special network service known as the Domain Name System (DNS) allows translation more user-friendly names (such as 'www.roxen.com') to the real, numeric address, so the friendlier names can be used when typing addresses.

The DNS approach gives the added bonus of being able to give several names to the same numeric address. The HTTP protocol takes advantage of this feature to allow several different virtual servers using the same port on the same machine by including a special 'Host:' header whenever a document is requested through the HTTP protocol connection. This is commonly referred to as 'virtual hosting'.

However, FTP has no way of doing this, and in HTTPS the 'Host:' header is theoretically available inside of the encrypted data, but unfortunately, the encryption and authentication strategies involved require that a particular host is selected before it has exchanged the keys necessary to decrypt the encrypted data containing the actual HTTP request, so HTTPS is also unable to use virtual hosting.

The URL (Uniform Resource Locator) that is used as the "visible" form of an address is a combination of a protocol name, an address (symbolic or numeric), an optional port number (if we're not using the standard port number for the protocol in question), and then optionally a

path inside the server. The first three parts are directly involved in how the client talks to the server, while the path is just a name of the page inside the particular server.

2.14.4 What it means for Roxen

The result of the situation described above is that there can only be one protocol per port, and that the HTTPS and FTP protocols are additionally restricted to one virtual host per port, while HTTP can have share one port between many virtual hosts.

Ports are configured for a server under the Settings->URLs in the Roxen configuration interface. Clicking 'New row' will make room for a new port, which is described by a simplified URL, which normally takes the form of protocol name followed by colon, followed by a double slash, followed by an asterisk (to indicate that we mean to use the local machine, whichever it is), and then optionally colon and a port number, if we don't mean to use the standard port number for the protocol.

For HTTP Roxen will automatically figure out whether any virtual hosting will be required (if you set up several HTTP services for the same port number).

2.15 Settings

2.15.1 UID

The UID variables control the user used to run Roxen under Unix. Usually services such as web servers are run as the root user, with unlimited privileges. However since Roxen might be running a lot of scripts and modules it is not always appropriate to run it as root.

There are three ways to run Roxen as another user. The simplest is just to start Roxen as that user. A problem is that only root may open privileged ports on many unixes. Therefore it might be necessary for Roxen to run as root to open the ports.

Then it becomes possible to use the Settings / Change uid and gid to variable. In that case Roxen will change to this user unless it has to do something that requires it to be root, for example opening a port. Since Roxen still retains the ability to change user to root this might not be safe enough. Therefore you can use the Settings / Change uid and gid permanently variable to force Roxen to permanently change user just after it has opened its ports. This is safer but has the side effect that you may have to restart the server to be able to change the URL settings of a site.

It must however be stressed that none of the methods of changing user id is as safe as starting Roxen with that user id. Paranoid persons solve the privileged port problem in another way, for example by putting a relay or load balancer in front of the web server or by using ACLs to give the Roxen user permission to open privileged ports.

2.15.2 Threads

On most operating system Roxen can run as a threaded server. This has the advantage of making it possible for Roxen to serve requests while it is doing something CPU intensive, such as creating a large business graphics diagram. Or serve other requests while it is waiting for the operating system to fetch a file from a slow disk.

It is however not necessary that threading makes Roxen perform better overall. There is always an overhead when running with threads. Roxen doesn't need threads to send data to several browsers at the same time, it only needs threads to be able to calculate what to send concurrently.

Threading will not make it possible for Roxen itself to make effective use of more than one CPU on a multi-CPU system.

It is possible to configure the number of request handling threads with the Settings / Number of threads to run variable. Note that modules and scripts may start more threads.

2.15.3 Modules Directories

The Settings / Module directories variable controls where Roxen will search for modules, by default in roxen/server/modules/ and in roxen/local/modules. The second location is intended for the user's own modules or third-party modules. It is not recommended to install any modules under roxen/server/modules, since they will be replaced when Roxen is updated.

Roxen will check any .pike, .class (Java modules) file found in the module directories or its sub directories to see if they are a module. A file is a module if it can be loaded and conforms to the Roxen module API.

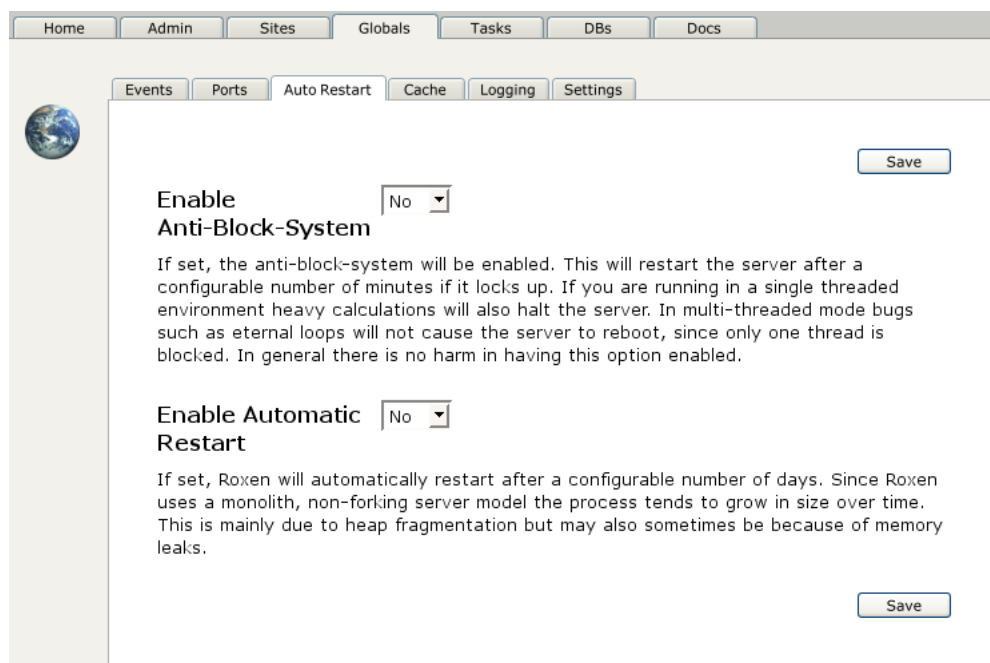
Note!

Module uniqueness is determined by the file name and only the file name. Roxen cannot handle two modules with the same file name, regardless of which directories they reside in. The upside to this is that you can move a module between directories without affecting Roxen.

2.15.4 Fonts

Roxen searches for fonts in the directories specified by Settings / Font directories, by default in roxen/server/nfonts and roxen/local/nfonts. The second location is intended for the user's own fonts or third party fonts. It is not recommended to install any fonts under roxen/server/nfonts since they will be replaced when Roxen is updated.

2.16 Auto Restart



The screenshot shows the Roxen Administration Interface. At the top, there are tabs: Home, Admin, Sites, Globals, Tasks, DBs, and Docs. Below these, there are sub-tabs: Events, Ports, Auto Restart, Cache, Logging, and Settings. The 'Auto Restart' sub-tab is selected. On the left side of the main content area, there is a small globe icon. The main content area contains two settings:

- Enable Anti-Block-System**: A dropdown menu is set to 'No'. Below it, a text block explains: "If set, the anti-block-system will be enabled. This will restart the server after a configurable number of minutes if it locks up. If you are running in a single threaded environment heavy calculations will also halt the server. In multi-threaded mode bugs such as eternal loops will not cause the server to reboot, since only one thread is blocked. In general there is no harm in having this option enabled." A 'Save' button is at the bottom right of this section.
- Enable Automatic Restart**: A dropdown menu is set to 'No'. Below it, a text block explains: "If set, Roxen will automatically restart after a configurable number of days. Since Roxen uses a monolith, non-forking server model the process tends to grow in size over time. This is mainly due to heap fragmentation but may also sometimes be because of memory leaks." A 'Save' button is at the bottom right of this section.

Globals / Auto Restart

2.16.1 Anti Block System

The Anti Block System restarts Roxen in case it locks up. It should not be necessary and is not enabled per default. But in case of an instable installation it can ensure that the server does not need manual intervention to be restarted.

2.16.2 Auto Restart

The screenshot shows a configuration window for 'Enable Automatic Restart'. At the top, there is a dropdown menu set to 'Yes' and a 'Restore default value' button. Below this, a text block explains: 'If set, Roxen will automatically restart after a configurable number of days. Since Roxen uses a monolith, non-forking server model the process tends to grow in size over time. This is mainly due to heap fragmentation but may also sometimes be because of memory leaks.' Under the 'Schedule' heading, there are three radio buttons: 'Never', 'Every 1 hour(s)', and 'Every 1 Day at 04 o'clock'. The 'Every 1 Day at 04 o'clock' option is selected. Below the schedule options, a note states: 'Automatically restart the server according to this schedule.' A 'Save' button is located at the bottom right of the panel.

Schedule

The auto restart system restarts Roxen once every nth day. Since Roxen is a very long lived process it is possible that the server grabs too much memory or other resources after having been up too long. The auto restart system makes it possible to ensure that this does not become a problem by periodically restarting the system.

To activate restart interval choose yes at "Enable Automatic Restart" and click save. Choose restart interval in the calender and click save again.

The auto restart system should not be needed and is not enabled per default. But it is there as a precaution.

2.17 Cache

Globals / Cache

The Globals / Cache tab contains settings for the argument, memory and proxy disk cache. The argument cache is a disk based cache used primarily by the image generating RXML tags. It is long lived since it is very hard to know how long clients will request one of the generated images. There is no automatic system for deleting information from the argument cache, the system administrator will have to do that manually or create a cron job to do it.

The memory cache is used for a number of things in Roxen, for example caching DNS queries, support entries or the cache tag. It is usually a fairly short lived cache, entries living a number of minutes. The Status / Cache status task shows information about how the memory cache is performing. The Maintenance / Flush caches task lets you clear the memory cache as well as the data cache.

The proxy disk cache is used by the HTTP proxy module, to store fetched files on disk. It is a long lived cache with its own garbage collector that removes old files and makes sure the cache stays within a certain size.

Apart from these caches there is also a data cache for frequently used static pages, which is described in the Sites chapter.

2.18 Modules

A module is an addition to a site, adding to or modifying the site's functionality in some manner. The module is made up of a Java or Pike object that is run inside Roxen. Each module is configured through the administration interface.

Which modules are enabled and how they are configured determines how the web site will behave, or even if it should be a web site at all. If only proxy modules are enabled the site will not be a web site but rather a proxy server.

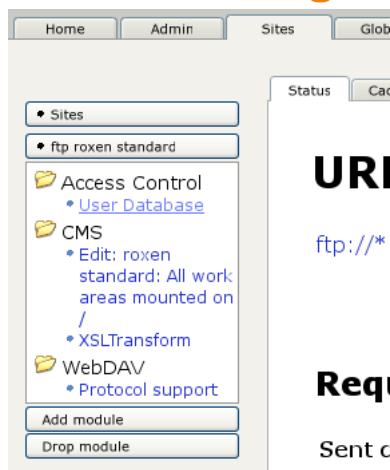
Modules come in different flavors, or types. Each module belongs to one or more types. Some types, like the Authentication and Directory types, are special and you can only enable one per site. It is on the other hand possible to enable any number of modules of more common types,

like the Location or Parser types. Some modules may themselves be installed more than once, but that is a module dependent property not something determined by the module type.

The module type determines what services the module provides. A Location module will provide files, either from a real file system, from a database or from some other source. A Parser module provides new RXML tags, that can be used like HTML tags.

The module types are designed so that modules can cooperate. The idea is that each module should provide a basic functionality, that can be combined by the functionality provided by other modules. That way the administrator can tailor the configuration of her web site to her needs, by choosing the right set of modules.

2.18.1 Installing



Adding a module.

In order to add a module to a virtual server press the Sites tab and then focus on the name of the site. As soon as you have done this you will see the Add module button at the bottom of the page. Pressing the Add module button will display a list of module categories. In each category is a list of available modules. Below each module header is a brief explanation of the module.

Some modules require other modules to work. When installing such a module the modules it requires will also be installed. It will not be possible to delete the supporting modules without first deleting the module that requires them.

Configuring the Module Path

Roxen searches for modules in the directories configured in the Module directories variable. It doesn't matter where you store your modules as long as the path to their directory is configured in this variable.

On the page for each module you can see where that particular module resides.

Upgrading a Module

To upgrade a module you must first replace the files for the old version with the files for the new version. Then you focus on the module and press the Reload Module button. If there are any problems with compiling the new version they will show up here. If so, you can always move the files for the old version back and no harm will be done.

Compilation Errors

If a module gets a compilation error this will be entered into the Event Log and the debug log.

2.18.2 Module Types

Almost all functionality in Roxen exists in different modules. Each module has a distinct task. Several modules can cooperate in the creation of a page that will be sent to the user.

What task a module has is determined by its type. More complex modules can be of several types and thus perform more than one task.

Authentication

modules handles authentication of, and information about, users. The most common type of Authentication modules are modules that import the user database from the operating system Roxen is running on. The information provided by an Authentication module is often used by other modules, such as the User Filesystem module. It is only possible to have one Authentication module per virtual server.

Directory

modules deals with directory listings and index files. If the requested resource is a directory, a directory module either tries to find a suitable index file or to create a page with a directory listing. It is necessary to have a Directory module in order to get index files, such as index.html, to work. It is only possible to have one Directory module per virtual server.

Extension

modules handle virtual files, with a certain extension. Each time a request is made to a URL ending with that extension, the extension module will be called. There are no Extension modules in the Roxen distribution.

File extension

modules deal with files with a particular extension, such as .html or .gif. The file in question must first have been delivered by a Location module. The ISMAP image-maps module is a File extension module.

Filter

modules filter data that is just about ready to be sent to the browser. This can be used, as the name suggests, to filter out parts of the data that should not be sent.

First try

modules are called before all other module types, except for Authentication modules. This is used to catch certain types of requests, for instance, to block access to your server from certain IP addresses or to send a warning message to the administrator if the server is accessed outside working hours.

Last try

modules are called when all other modules have failed to produce anything at all from the request. A Last try module could give an elaborate error message.

Location

modules deal with file systems, fetching files and directories. A Location module could work with a real file system or a purely virtual one. For example, it could fetch files from a database instead of a file system.

Most web applications are also implemented as Location modules. Pike and CGI scripts work much like a Location module.

Each Location module is mounted somewhere on Roxen's virtual filesystem. Several Location modules may be mounted on overlapping mount points. If the module have the same priority, the module with the longest mount point will be called first. Thus a module mounted on `/schedule/server/` will be called before a module mounted on `/schedule/`.

Logging

modules perform some logging of information about the requests. This could be done by writing log files or in some other way. The Logging module decides whether the request has been logged properly or whether the request should also be logged by the built-in log system.

The Main parser

module handles all RXML parsing. The module handles the interface to Parser modules. A Main parser must be installed for any RXML parsing to take place.

The Main parser must somehow get pages to parse. This is usually done by making the module a File extension module as well. Thus all files with a certain extension will be parsed.

There can only be one Main parser module in each virtual server. The RXML 2.0 parser module is the only Main parser module included in the Roxen distribution.

Parser

modules define one or more RXML tag. This is one of the most common user created modules. Making new tags available is an excellent way of making functionality available to the users of the server.

The Parser modules are called upon by a Main parser module. If no Main parser module is installed in the virtual server, no RXML parsing will take place.

Protocol

modules set the protocols the virtual servers can use. It handles a network connection to a port and then sends the request on to Roxen. HTTP, HTTPS and FTP are examples of protocol modules included in the Roxen distribution.

Provider

modules provides other modules with different services and extra functionality. They do not in themselves have anything to do with the normal request handling.

Proxy Types

module sets the content type of a file, if it hasn't already been done by the preceding modules. This is usually done by looking up the Mime type for a certain file extension in a database. There can only be one Types module in each virtual server. The Content Types is the only Types module in the Roxen distribution.

URL

modules receives one URL and returns another. In other words, the URL modules transform requests into other requests. This is useful when a web page is published under several names, or when a web page has moved.

2.18.3 File Systems

To make files available via a Roxen server there are a few different types of modules which cooperate.

The source of all files are one or more Location modules. They provide files as well as directory information. Location modules can be mounted on overlapping mount points, in which case the union of the files provided by the modules will become available.

A Directory module is used to handle index files, such as index.html, or produce directory listings.

File extension modules handle files of a certain extension. The Main RXML parser module is a file extension module that usually handles .html or .rxml files.

If no File extension module handles a certain extension the Content types module will be called to determine the appropriate content-type for a file.

2.19 Scripting

One of the most exciting things about the web is that you can make your own applications, that will be reachable by anyone in the world. Furthermore, programming for the web is often simpler than traditional GUI programming. Even small applications can get nice graphical user interfaces by creating dynamic HTML pages. Roxen is one of the best environments for creating such applications.

As with all good things, there are drawbacks. Since an application on the web is reachable by any number of users, some with malicious intent, programming errors can have drastic effects. While many users may not understand this, the administrator of a web server must.

The important thing is that all user input must be handled with caution. Where the programmer thought he would get a small name he might get ten megabytes of machine code. If the program fails to handle that kind of input, troubles might follow.

Building web applications with Roxen in Pike reduces the risks and consequences of making such mistakes, but it does in no way eliminate them.

Roxen also supports CGI scripts for doing scripting. It is far easier to make fatal mistakes when programming CGI scripts than it is with Pike scripts or modules. Most CGI scripts that can be downloaded from the web have not been written with security in mind. As system administrator, you must determine which scripts are safe and which ones are not, and consider your site's security policy.

It is always a good idea to keep track of Roxen's log files. If outside users try to break in through CGI scripts, it will most often show up in the log files. Especially since they will usually try to break in through a few common CGI scripts.

This chapter describes Roxen's various ways of supporting script programming from a system administrator's viewpoint.

2.19.1 CGI

CGI scripts are the most common way of doing scripting, being supported by virtually every web server. They work by starting an external program for each request. The program can be a compiled program or a script written in something else, for example like perl or python.

The advantages of CGI scripts are that they can be used to run any kind of scripts written in any language, and that they are portable between different web servers. The disadvantages are that

they are resource hungry, needing to start an external program for each request. CGI scripts are also one of the most common security hazards on the web. Most often because they were not written with security in mind.

In Roxen CMS, CGI scripts are supported through the CGI executable support. The module can be configured to either run the CGI scripts as the user who has written them, or to run all CGI scripts as a user with low privileges.

It is possible to use CGI scripts together with RXML. Either by letting the output of the CGI script be parsed with the RXML parser or by executing the CGI scripts with the `<cgi>` tag.

Even if CGI scripts are run so they cannot hurt the server itself, they can often hurt the user who owns them. Most users do not understand how CGI scripts work, but will download and install them from the net, with no thought of security. It is often better to provide the functions the user's want as RXML tags. The user will understand RXML tags better, since they are like HTML tags, and the administrator will get better control over the server.

2.19.2 Fast CGI

Fast CGI is a more efficient way of doing things than regular CGI scripts. Fast CGI works by launching an external process which keeps running once started, processing one request after another, instead of having a new process started for each request.

Roxen's FastCGI support is handled through the Fast CGI scripting support module.

As with CGI scripts, there are security hazards involved with running Fast CGI scripts. They are most commonly caused by the low quality of existing Fast CGI scripts.

2.19.3 SSI

SSI is short for Server Side Include, and is essentially a way of running CGI scripts within HTML pages. They are like RXML tags, but not as flexible. SSI is a standard that works on several web servers.

SSI support is handled through the Main RXML parser and is turned off by default.

SSI works like a simple pre-processor for HTML, and reads directives that are hidden in HTML comments. One directive, `<!--#exec-->`, allows a user to execute programs and is thus a potential security hazard. The programs will be executed as a user with few privileges, by default nobody. The directive can also be disabled altogether.

2.19.4 Java servlets

Java Servlets is a new form of scripting that is gaining popularity because it allows portable server extensions that do not rely on starting external binaries. Servlets are written in Java and compiled to Java byte code before they can be used in a web server.

In Roxen, Servlets are supported through the Java servlet bridge module, which transforms a Servlet into a Location Module. The Java servlet bridge module can have any number of copies, so any number of Servlets can be installed on a single virtual server.

To run the Servlet support you need a Pike that has been compiled with Java support. The binary versions of Roxen CMS are not compiled for Java support. To compile a Pike with Java support you have to install JDK and then compile Pike. Pike's Java module should detect the Java environment automatically.

2.19.5 Roxen modules

Roxen modules are, strictly speaking, not scripts in the same sense as the other script types. Rather, they are a way of extending the server's functionality. Most of Roxen's functions are implemented as modules. However, since modules are often a convenient way of doing things that might otherwise have been done with regular scripts, modules belong in this overview.

Roxen modules can be written in Java or Pike.

Roxen modules can be used to implement new RXML tags, support new communication protocols, and numerous other things that would be impossible or difficult with other script types. Modules are also persistent entities in the server, as opposed to ordinary CGI scripts, which are executed in a single-shot fashion, starting anew each time they are invoked. FastCGI scripts are somewhat persistent.

From a security viewpoint, Roxen modules must be treated with utmost care. Modules have access to the whole server. Thus, only trusted users should be able to write their own module, and modules should only be downloaded from reputable web sites. Apart from the security considerations, modules that take a long time executing can also cause performance degradation and other service disruptions, especially if the server isn't running with enough threads.

Compile time and run time error messages for modules will be reported, with a Pike backtrace so the problem can be pinpointed. All such error messages will be reported on the Event Log tab and in the debug log. They will also be reported on the web page where the error occurred, unless the Show the internals variable is disabled.

2.19.6 Pike scripts

Pike scripts are an easy and quick way of doing scripting in Roxen. Since Pike is also the language that Roxen uses internally, Pike scripts are efficient, and easy to handle.

Support for Pike scripts in Roxen is provided by the the Pike script support module.

As with Pike modules, error messages are reported complete with a Pike backtrace. This makes development fast since it is simple to pinpoint the problem.

Pike scripts are usually persistent. That means they will be compiled only once, and invoked repeatedly. This makes them fast and efficient.

2.19.7 Pike tag

The Pike tag is an RXML tag, allowing the user to insert bits of Pike code into RXML pages. This is very convenient, if the function needed exists as a RXML tag, use it, otherwise, write a small Pike function that solves the problem. Unfortunately it is also dangerous. The Pike code within the tag has full access to the server. Therefore, it is important that only trusted users have access to the Pike tag. It is also very important to make sure that all user input is quoted properly before being inserted in a web page.

As with all Pike development in Roxen CMS, error messages will be reported together with a backtrace, making it easy to pinpoint any problems.

2.19.8 Perl support

The Perl support provides support for both Perl scripts and Perl tags. In both cases, the actual Perl code is executed in a subprocess, and not in the main Roxen process, which makes it harder for the programmer to accidentally mess things up for the main Roxen server.

The module will catch requests to documents whose path ends with any of the extensions (suffixes) indicated by the 'Extensions' setting in the module configuration. The default is '.perl' and '.pl'.

Perl tags, or processing instructions, will be available if the module setting 'Enable Perl tag' is set to 'Yes'. This means ordinary RXML pages can use Perl code, such as in the example below.

```
<?perl print scalar(localtime), "\n"; ?>
```

It is possible to enable similar parsing of script results by switching the 'Script output' option to 'RXML', but this is usually not desirable, and may not work so well with traditional CGI scripts. Nevertheless, it might be useful under special circumstances.

When configuring the Perl support, it may be worth considering how heavy load the Perl support will be expected to handle. If a Roxen site makes heavy use of Perl scripts, it may be desirable that several such scripts can execute in parallel, while the opposite may be true if your Roxen server runs on modest hardware. In either case, the maximum number of subprocesses on behalf of the Perl support can be decided through the 'Parallel scripts' setting in the module configuration. The default is 3, which should be good enough in most cases, but it might be desirable to raise or lower it some cases.

Unless other arrangements are made by the scripts, errors occurring during the execution of Perl scripts will be reported in the server debug log, so this is the place to watch for error reports. (The Event log in the web interface won't show these errors.)

Consult the Roxen Programmer manual for more details about how to use the Perl support.

2.20 Performance: Average Profiling

Average profiling is a profiling tool for RXML applications. The tool counts calls to tags and modules and accumulates the time spent on each call. The registered data can later be viewed with a web based user interface.

To enable average profiling the webserver has to be started with the -DAVERAGE_PROFILING flag. If the flag is given, profiling will be enabled and an extra module in the administration interface will be loaded. The extra module, Average Profiling, provides some task buttons and a web based interface for the profiling data.

New profiling data will be stored in the webserver's memory until the Dump to database button is pressed. Then the data will be stored in an SQL database. Every click on the dump button will create a new session.

2.20.1 Average Profiling Module

The settings tab have these options:

Location

The mount point for the user interface.

Database

The database to store and read information from.

The information tab has some tasks buttons, they are:

Clear database

Clear the entire database table.

Dump to database

Save new profiling information to a database in a new session.

Clear profiling information

Clear new profiling information from memory.

2.20.2 User Interface

The user interface is reached by clicking on the link on the modules information page. The interface consists of two parts; the selection and the result part. The following selections are available:

Session

Select events for a special session. Each time the dump to database button is pressed a new session will be created.

Configuration

Select events for a specific virtual server.

File Glob

List only those files that matches this glob pattern.

Event Class

Select events with a specific event class.

Event Name

Select events with a specific event name. This select box can contain lots of entries. If the event class is selected first, only the event names in that class will be listed.

Sort by

Select what column to sort by.

Direction

Select the sort direction.

Group by

Select what column to group by. It is possible to group by file, event class and event name. Grouping is a technique to combine entries with common values on the group by column. Numerical columns are accumulated and textual columns are hidden.

An event is something that has been timed, for example to generate the result from a tag. An event belongs to an event class, for example the event gtext belongs to the class tag. The event classes and their content is listed below:

Event Class	Generated From	Example Event Name
Directory	Directory Modules	Directory Listings
Emit	Emit Plugins	dir
Entity	RXML Entities	dir.title
Ext	Extension Modules	Scripting: Pike script support

Filter	Filter Modules	Tags: Javascript Support
Location	Location Modules	SiteBuilder: Content Editor
Overhead	Various overhead	rxml
Tag	Tag Modules	gtext

The result part of the user interface consists of a table containing the matched entries. A description of each column is given below:

Session

The session.

Configuration

The virtual server name.

File

The file the event occurred in.

Event Class

The event class.

Event Name

The event name.

Calls

The number of calls to the event.

Real (ms)

The total real time in milliseconds for the calls.

Av. Real (ms)

The average real time in milliseconds for each call.

CPU (ms)

The total CPU time in milliseconds for the calls.

Av. CPU (ms)

The average CPU time in milliseconds for each call.

3 Appendix A: Glob Expressions

A glob pattern, is a sort of string that can be compared and matched against several other strings. The rules for a glob pattern are fairly simple. There exist two special characters, "*" and "?". "*" can match zero or more characters of any kind while "?" can match one, and only one, character of any kind. All other characters match themselves. Some examples:

"Hello" matches "Hello" and only "Hello", e.g. not "hello".

"Hi?" matches all string with three letters than begin with "Hi", e.g. "Him" and "His", but not "Hi" nor "Hiss".

"Hi*m" matches all strings that begin with "Hi" and ends with "m", e.g. "Him" and "Hi mum".

"Hi?*m" matches all strings that begin with "Hi", ends with "m" and has at least one character between "Hi" and "m".

"Hi*m*" matches all strings that begin with "Hi" and has an "m" in it.

Note!

It is not possible to escape the characters "*" and "?" so that they match themselves. If you need to do such a match you have to use a regular expression.

4 Appendix B: Regular Expressions

Regular Expression, often abbreviated regexp, is a standardized way to make patterns that match certain strings. Normal characters, such as A through Z only match themselves, but some characters have special meaning.

Pattern	Matches
.	Any one character.
[abc]	a, b or c.
[a-z]	Any character a to z inclusive.
[^ac]	Any character except a and c.
(x)	x, where x might be any regexp.
x*	Zero or more occurrences of x, where x may be any regexp.
x+	One or more occurrences of x, where x may be any regexp.
xly	x or y, where x and y may be any regexp.
xy	xy, where x and y may be any regexp.
^	Beginning of string, but no characters.
\$	End of string, but no characters.
\<	The beginning of a word, but no characters.
\>	end of a word, but no characters.

Some examples:

Regex	Matches
[0-9]+	One or more numeric characters.
(foo) (bar)	Either "foo" or "bar".
\.html\$	Any string ending in ".html".
^\.	Any string starting with a period.

Note!

\ can be used to quote these characters in which case they match themselves, nothing else.