

Table of Contents

Installation	3
Software Installation	3
Creating a Site	8
Upgrading a 1.3 Site	9
Java Support.....	10
Databases.....	10
Updating	13
Manual Update.....	13
Update Server	13
Administration Interface	15
Admin Tab.....	15
Sites Tab	17
Globals Tab	18
Ports Tab	18
Events Tab.....	18
Tasks Tab.....	19
Docs Tab	19
Sites	21
Site Information.....	21
Site Settings	21
Logs	23
Event Log	23
Debug Log	23
Access log	24
Accessed Counter	24
Globals	27
Settings	27
ABS	27
Auto Restart.....	28
Cache	28
Modules	29
Installing	29
Module Types	29
File Systems.....	30

Table of Contents

Installation

To install a Roxen server 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 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 of Roxen relies on code outside of Roxen 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 the Roxen server itself.

Installing the Software

The Roxen server consists of a Pike interpreter and the Roxen server 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.

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.

Sites

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

Upgrading

Upgrading from Roxen 2.0 to Roxen 2.1 should pose no major problems. Upgrading from Roxen Challenger 1.3 or older to Roxen 2.1 is harder, since there were some major changes introduced in the 2.0 release. However there are modules available that make Roxen 2.1 backwards compatible with Roxen Challenger 1.3. Old sites are supported with no changes to the site content. Unfortunately the way the configurations are stored has changed as well, so you have to configure your sites manually rather than only upgrading the new software.

Migrating

Moving an existing web site to a Roxen server should pose no major problems, if Roxen supports all scripts that are used on the site. Usually you only have to tell Roxen where the web site is located. If scripts are used you also need to enable modules to handle the scripts.

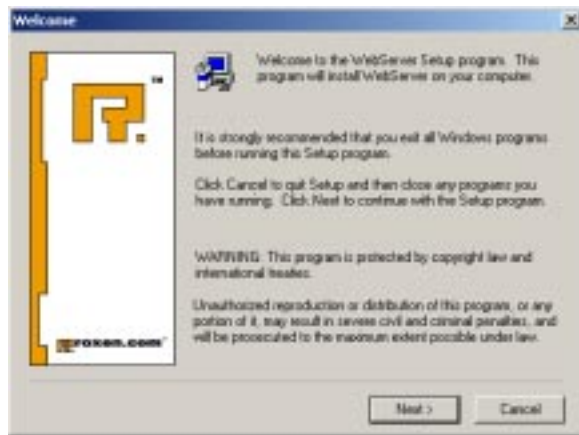
End of /roxen/2.1/administrator/installing/index.xml

Software Installation

Start of /roxen/2.1/administrator/installing/software.xml

The software installation installs Pike and Roxen 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.

Windows Binary Installation



Installation wizard

The Windows binary release is distributed as a self-installable package. Just start it and a wizard will guide you through the installation.



Administration interface dialog

The first dialog with actual Roxen specific settings is for creating the administration interface for your server. The settings are *Server name*, *Protocol*, *Port*, *Admin user name* and *Admin password*. *Server name* is the name of the administration interface site.

Protocol is the protocol that will be used to access it. It should normally be *https* to ensure that the traffic is encrypted. Note that it is also necessary to create a new certificate, since it will use the insecure demo certificate by default.

Admin user name and *Admin password* are the user name and password who will be used to access the admin interface.



Update client dialog

The next dialog asks whether your Roxen server will connect to the update server at roxen.com automatically or not. If you allow it the builtin update client will fetch information about new software and updates as well as news about Roxen. The information will be displayed in the *Admin* tab of the administration interface.

If you do not allow this it is still possible to use the update client, it will connect to the update server and fetch information when you tell it to. The disadvantage is that you will not get automatic alerts about new security fixes and other important news.

If you have a Roxen Community user you can let the update client connect with your user name and password, to let it download packages that require authentication. The Roxen Community, <http://community.roxen.com/>, is a community for Roxen users. The update server is part of Roxen Community.



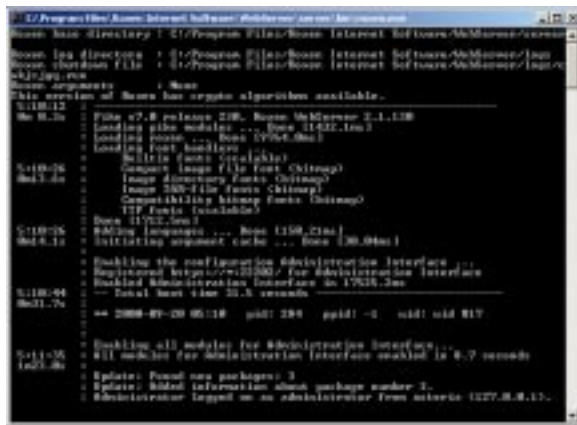
Install as a service dialog

If you install on Windows NT or Windows 2000 you will also be prompted whether you want to run Roxen as a service or not. If you are installing to run the web server in production you should run it as a service. If you are installing to do development it is better to start it manually from the *Start* menu.



Last dialog

The last dialog will enable you to start Roxen. If you do so Roxen will be started with the debug log in a console windows.



Roxen console

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-2.0.[build]-[os].sh
```

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 will be installed in `/usr/local/roxen/`. Roxen 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.

Next it will ask whether to want to install Pike separately or only as a part of Roxen. Installing Pike separately is recommended since it will make it possible to run other Pike programs as well. If you answer yes you will be prompted for the location of the Pike installation.

Install Pike separately [Y/n)?

Pike installation prefix: `/usr/local/`

Now the actual installation of the software will take place. This takes a while and shows nice progress bars. Next the installation script will 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
/home/wing/cto/manual/roxen/configurations/.
```

Server name: Administration Interface

Port URL: `http://*:27474/`

Administrator user name: administrator

Administrator password:

Administrator password (again):

Are the settings above correct [Y/n]?

The last stage of the installation process asks whether your Roxen server will connect to the update server at `roxen.com` automatically or not. If you allow it the builtin update client will fetch information about new software and updates as well as news about Roxen. The information will be displayed in the *Admin* tab of the administration interface.

If you do not allow this it is still possible to use the update client. It will then connect to the update server and fetch information when you tell it to. The disadvantage is that you will not get automatic alerts about new security fixes and other important news.

If you have a Roxen Community user you can let the update client connect with your user name and password, to let it download packages that require authentication. The Roxen Community, `http://community.roxen.com/`, is a community for Roxen users. The update server is part of Roxen Community.

```
Roxen has a built-
in update system. If enabled it will periodically
contact update servers at Roxen Internet Softwar
e over the Internet.
```

Do you want to enable this [Y/n]? `Y`

```
If you have a registered user identity at Roxen
Community
(http://
community.roxen.com), you may be able to access add
itional
material through the update system.
```

```
Press enter to skip this.
```

Roxen Community identity (your e-mail): `wing@roxen.com`

Roxen Community password:

Roxen Community password (again):

Do you want to access the update server through an HTTP proxy [y/N]?

The last question is for users who run their Roxen server behind a firewall. Then it might be necessary for the server to use a proxy like any web browser between a firewall.

With this the installation script has all the information it needs. It will finish the installation and start the Roxen server. Further configuration is done by connecting to the administra-

tion 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.
```

```
The initial installation is complete. Starting R
oxen WebServer,
please wait.....
```

```
Congratulations! Your webserver has started. You
may now proceed with the
installation from the administration interface u
sing a web browser:
```

```
http://holt:27474/
```

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 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, PHP4 or GTK/GNOME support is 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. If you later find out that some function is lacking because of a missing library you will have to reinstall Roxen, 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 and upgrade rather than overwrite it.

The installation process consists of four steps; configuration, compilation, software installation and Roxen installation.

```
./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... ye
s

```

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="$SPIKE_PATH_TRANSLATE"; if test "x$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

```

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

Upgrading to a newer 2.x 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'

```

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://*:22424/
Administrator user name: administrator
Administrator password:
Administrator password (again):

Are the settings above correct [Y/n]?

```

Roxen has a built-
in update system. If enabled it will periodically
contact update servers at Roxen Internet Softwar
e over the Internet.

```

Do you want to enable this [Y/n]?

```

If you have a registered user identity at Roxen
Community
(http://
community.roxen.com), you may be able to access add
itional
material through the update system.

```

Press enter to skip this.

Roxen Community identity (your e-mail): wing@roxen.com
Roxen Community password:
Roxen Community password (again):
Do you want to access the update server through an HTTP proxy [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 Administrati
on Interface ...
: Registered http://*:27474/
for Administration Interface
: Enabled Administration Interface in 412
.lms
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
```

End of `/roxen/2.1/administrator/installing/software.xml`

Creating a Site

Start of `/roxen/2.1/administrator/installing/create-standard.xml`



Sites tab

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



Create new site

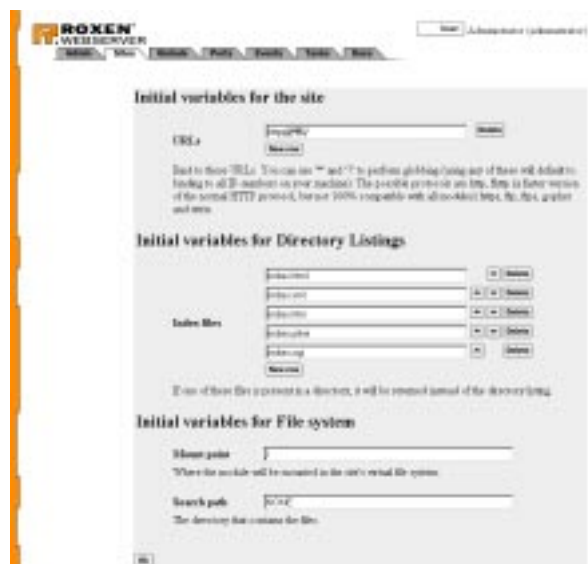
The first step is to give a name to the site. The name is only used to identify the site in the administration interface, it will not be exposed externally.

Press *Create new site with template* to continue.



Site template

Choose *Standard site*.



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, HTTPS and FHTTP. FHTTP is not really a protocol but rather another implementation of the HTTP protocol. It is significantly faster than the normal HTTP implementation, but does not support as many features. Because of this some modules will not work with FHTTP.

If you want to use another protocol you just change the URL accordingly: `ftp://your-domain/`, `https://your-domain/` or `fhttp://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 *Filesystem* 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 *Filesystem* 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.

End of `/roxen/2.1/administrator/installing/create-standard.xml`

Upgrading a 1.3 Site

Start of `/roxen/2.1/administrator/installing/upgrade.xml`

The 2.0 release introduced major changes to RXML, tag modules, the port handling as well as the administration interface. However most of the changes are backwards compatible, it is possible to move a 1.3 site to 2.0 with no or very minor changes.

The configuration files have changed, mostly because the ports has moved from the sites (or virtual servers) to being top level object. It is therefore not possible to just upgrade an old site, you need to reinstall it. The same, or compatible, modules are available.

RXML

Roxen 2.0 uses a new XML compatible RXML parser, which means that RXML tags must now always be terminated with an end tag or a `/`. For example `<imgs />`. Output tags, (formoutput, sqloutput, ldapoutput, etc) have been replaced with the `<emit>` tag and entities. Many simple tags have been replaced with entities instead. `<realfile>` is now `.`

The new RXML is simpler to use and less prone to errors. Especially the more complex uses of RXML, such as databases, are much simpler and safer. It is therefore important to convert existing sites to RXML 2.0.

However it is not necessary. By enabling the *Old RXML compatibility* module Roxen 2.0 will be compatible with the old RXML parser. This compatibility mode will be kept as long as it is needed.

Gradual Upgrade

Since the changes to RXML are so great it might be necessary to gradually move from RXML 1.3 to RXML 2.0. This can be achieved in several ways. The safest way is to keep the 1.3 release running in parallel with a 2.0 server, and switch when everything works. By using the *HTTP relay* module the web site can still seem like one, although two servers handle it.

Another way to handle the gradual upgrade is to use two sites: one new RXML 2.0 site and one with the *Old RXML compatibility* module installed. The compatibility site could be run on an URL of the new site, for example `http://my-site/old/`. By using the *HTTP redirect* module it is then possible to overload the two sites so that it seems like one site.

Platform

A 2.0 SiteBuilder uses the same storage as a 1.3. Upgrading should be a matter of installing a new Platform server and choosing the old storage. However, since the 1.3 release will not work with a storage that has been used by the 2.0 release, it is smart to make a copy of the storage before upgrading.

As with the rest of RXML the SiteBuilder specific tags have changed. You need the compatibility module to get the old behavior. Fortunately most SiteBuilder specific RXML should be located in the template files, which limits the migration work.

The template system has changed, as well as how templates are found. We use XSLT templates rather than the old template module. It is still possible to use old style templates but they should be phased out. It is possible to make XSLT templates that emulate any behavior of old templates, it is not necessary to change the data files. The new XSLT template processor can work with HTML as input format. However if you have defined any singletags of your own, in the templates, it is necessary to change the data files from `<singletag>` to `<singletag />`.

Modules and Pike Scripts

The module API is backwards compatible with the 1.3 release. Minor differences in the inheritance as well as in Pike's class library might cause compatibility problems with a few modules or scripts. If you site relies on third-party modules or script you should wait until they have been tested with the 2.0 release. If you have programmed any modules or scripts yourself you should test them on a 2.0 server before upgrading. The compatibility errors should be found when compiling the modules or scripts and thus be easy to locate.

End of `/roxen/2.1/administrator/installing/upgrade.xml`

Java Support

Start of /roxen/2.1/administrator/installing/java.xml

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.

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.



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.



Java servlet bridge information

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.

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.

End of /roxen/2.1/administrator/installing/java.xml

Databases

Start of /roxen/2.1/administrator/installing/databases.xml

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@mysql-host.roxen.com/test`.

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



SQL tag module

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



Connected

If everything does not work the information tab will give an error message. For example if could not find the correct driver.



No driver found

Or if it could connect to the database server, but the user name, password or database was wrong.



Wrong password

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.



Pike module list

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*.

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.

Msql

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

Postgres

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

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
```

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`.

Oracle

Oracle is supported through a native driver.

Sybase

Sybase is supported through a native driver.

Informix

Informix is supported through a native driver.

Mimer

Mimer is supported through its ODBC driver.

End of /roxen/2.1/administrator/installing/databases.xml

Updating

Manual update

Manually updating your Roxen is in most cases very similar to installing it. The Windows versions demand a complete reinstallation, i.e. all files will be overwritten except for configuration files and files created by users. The various Unix tastes only replaces the `server` directory with a new one by renaming the old directory `server-old`.

Update server

The *Update Server* is not operational yet.

End of `/roxen/2.1/administrator/updating/index.xml`

Manual Update

Start of `/roxen/2.1/administrator/updating/manual_update.xml`

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

Windows Binary Update

Updating 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 be installed anew, by overwriting all the old files. 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, else the information will be lost.

Unix Binary Update

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-2.1.[build]-[os].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-2.1.[build]/`. The first time Roxen is installed this directory is called `roxen/server/`. This directory will be renamed to `roxen/server-old/` when Roxen has been updated.

Install Pike separately [Y/n]?

Pike installation prefix: `/usr/local/`

Enter the path to your current Pike installation. Pike will be installed in a versioned directory, e.g. `pike/7.0.[build]/`.

Take care not to keep any essential code in your old `roxen/server/` directories. After three updates 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.

Unix Source Update

The result when updating 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.

End of `/roxen/2.1/administrator/updating/manual_update.xml`

Update Server

Start of `/roxen/2.1/administrator/updating/update_server.xml`

Update server

Currently the update server is being redesigned, hence it is not possible to update your Roxen installation through it.

End of `/roxen/2.1/administrator/updating/update_server.xml`

Administration Interface

Configuration and maintenance of Roxen is handled by the web-based administration interface. The administrator connects to the interface with a browser and can then access all configuration options and maintenance tasks.

Tabs

The administration interface consists of seven tabs; *Admin*, *Sites*, *Globals*, *Ports*, *Events*, *Tasks* and *Docs*. The *Admin* tab contains settings for the administration interface itself as well as the update client that can download new versions of Roxen and additional software.

The *Sites* tab contains the web- and ftp sites handled by the Roxen server, as well as all module configurations. Most of the administration work is done on the *Sites* tab.

The *Globals* tab contains configuration settings affecting the whole Roxen server, such as where to find modules or which UID to run under. The *Ports* tab contains all port settings, that is the http or ftp ports used by the web- or ftp sites. Since several sites can share the same port the port settings are located on a separate tab.

The *Event* tab contains the event log, a web based version of the debug log. Error messages for the entire web server will be written here. Error messages relating to a certain site or module are also written to the site or module page.

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 or getting debug status.

The *Docs* tab contains the online documentation.

Administration Interface Site

The administration interface is in fact a site almost like any other web site handled by Roxen. It is implemented using RXML and a few custom modules. It is however not recommended to do any changes to the module settings of the administration interface site. The only setting that can be changed is the URL.

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.

End of `/roxen/2.1/administrator/admin-interface/index.xml`

Admin Tab

Start of `/roxen/2.1/administrator/admin-interface/admin.xml`

The administration interface settings are located at the *Admin* tab as well as the *Administration Interface* site under the *Sites* tab. Even though it is possible to change any settings in the *Administration Interface* site, only the *URLs* variable should ever be changed. Changes to the *Administration Interface* site can make it impossible to configure your Roxen server.

The *Admin* tab contains four tabs; *Startpage*, *Update*, *Your settings* and *Users*. *Startpage* contains a welcome message and news downloaded by the update client.

Your settings contains the settings for this administration interface user. These settings control what information is shown in the interface, as well as its layout.

Users contains the accounts for all users who have access to the administration interface as well as their security privileges.

Creating an Administration Interface

The administration interface is created as part of the installation process described on the Software installation page of the Installing chapter. It is also possible to run the administration interface creation script, `roxen/server/create_configinterface`, manually.

Site Settings



Administration interface site settings

It is possible to change all site settings of the administration interface site. However the only setting that should ever be changed is the site URL or URLs. By default the site is mounted on its own port. It can however be hard to remember the port number, and hard to reach the port through a firewall. Sometimes this is a good thing, sometimes it makes the administration harder.

One way would be to create a whole new web site for the administration interface, `https://admin.company.com/`. But it might be too much trouble to get an IP number for the administration interface (remember a https port needs its own IP number). Another way is to mount the administration interface on the web site itself. For example `https://www.company.com/admin/`.

When changing the URLs of the admin interface, don't remove the old URL until you've made sure that the new one works. Otherwise you might not be able to configure your Roxen server anymore. If that should happen you might need to

restore the settings from a backup file, or edit the configuration files manually.

The configuration file for the admin interface is usually `roxen/configurations/Administration_Interface`. The most recent backup file is `roxen/configurations/Administration_Interface~`. After the configuration interface settings are restored, restart the server by killing its process. You can get its process id by running `grep pid roxen/logs/debug/default.1`.

In Windows you would instead first stop the Roxen service and then start it again, kill the Roxen console and run Roxen again, or end the Roxen process using the Task Manager.

```
holt manual $ grep pid roxen/logs/debug/default.1
: ** 2000-09-
20 07:54 pid: 15469 ppid: 15460 uid: wing
8:00:00 : ** 2000-09-
20 08:00 pid: 15469 ppid: 15460 uid: wing
holt manual $ kill 15469
holt manual $
```

Users



Admin / Users tab

Several users can have access to the administration interface, with their own password and different privileges. The users are administrated on the *Users* tab. Here it is possible to create new users, change a users password or change her privileges.

To log in as another user click on the user button in the top right corner.

Your Settings



Admin / Your settings tab

Each administration interface user has a number of configurable settings that control the layout of the administration interface. It is possible to customize how the information in key parts of the interface, such as the module list, is presented. It is also possible to change theme, to get a different layout.

It is not possible to change the settings for another user. To do that you first have to log on as that user by clicking on the user button in the top right corner.



Administration interface with new theme

End of `/roxen/2.1/administrator/admin-interface/admin.xml`

Sites Tab

Start of `/roxen/2.1/administrator/admin-interface/sites.xml`

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 focus on 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.

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.

Module Page



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.

End of `/roxen/2.1/administrator/admin-interface/sites.xml`

Globals Tab

Start of `/roxen/2.1/administrator/admin-interface/globals.xml`



Globals tab

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

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

End of `/roxen/2.1/administrator/admin-interface/globals.xml`

Ports Tab

Start of `/roxen/2.1/administrator/admin-interface/ports.xml`



Ports tab

The *Ports* tab contains a list of all ports this Roxen server handles. For each port the URL is listed as well as which sites use the port. By clicking on the port you focus on that port and can change its settings. By clicking on the site you come to that site. The ports are described in more detail in the Ports chapter.



Port page

The port page contain the settings of the port, the actual settings differ with the protocol used. This is described in more detail in the Ports chapter.

End of `/roxen/2.1/administrator/admin-interface/ports.xml`

Events Tab

Start of `/roxen/2.1/administrator/admin-interface/events.xml`

End of /roxen/2.1/administrator/admin-interface/docs.xml



Events tab

End of /roxen/2.1/administrator/admin-interface/events.xml

Tasks Tab

Start of /roxen/2.1/administrator/admin-interface/tasks.xml



Tasks tab

End of /roxen/2.1/administrator/admin-interface/tasks.xml

Docs Tab

Start of /roxen/2.1/administrator/admin-interface/docs.xml



Docs tab

Sites

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.

Create a New Site

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

Delete a site



Drop site

To delete a site press the *Drop old site* button and choose which site to delete.

End of </roxen/2.1/administrator/sites/index.xml>

Site Information

Start of </roxen/2.1/administrator/sites/site.xml>



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.

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.

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.

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.

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.

End of </roxen/2.1/administrator/sites/site.xml>

Site Settings

Start of </roxen/2.1/administrator/sites/settings.xml>



Site settings

The site settings are divided into four tabs: *Cache*, *Logging*, *Settings* and *Throttling*. The most important settings are under *Settings*. *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.

Note! Some site related settings are also done under the *Port* tab and some settings done in various modules could be considered being site settings.

Settings

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 *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.

Default site controls which site will be chosen if there are several to chose 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.



404 Page not found

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.

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.

End of /roxen/2.1/administrator/sites/settings.xml

Logs

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.

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.

End of `/roxen/2.1/administrator/logs/index.xml`

Event Log

Start of `/roxen/2.1/administrator/logs/event-log.xml`



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



Event log in module page

End of `/roxen/2.1/administrator/logs/event-log.xml`

Debug Log

Start of `/roxen/2.1/administrator/logs/debug-log.xml`

```
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

```

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.

End of `/roxen/2.1/administrator/logs/debug-log.xml`

Access log

Start of `/roxen/2.1/administrator/logs/access-log.xml`

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

```

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.



Globals / Logging

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



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.

End of `/roxen/2.1/administrator/logs/access-log.xml`

Accessed Counter

Start of `/roxen/2.1/administrator/logs/accessed.xml`



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.

End of `/roxen/2.1/administrator/logs/accessed.xml`

Globals

The *Globals* tab contains settings that affect the whole web server. It consists of five tabs; *ABS*, *Auto Restart*, *Cache*, *Logging* and *Settings*. *ABS*, auto break system, can restart the web server in case it hangs. *Auto Restart* schedules automatic restarts of the server. *Cache* controls the memory, argument and proxy disc cache. *Logging* controls logging for the debug and access logs, and is described in the *Logs* chapter. *Settings* contains lots of stuff.

Some global settings are not configurable through the *Globals* tab but must be sent as options to the start script. See the foo appendix for information about the options to the start script, or run `start --help`.

End of /roxen/2.1/administrator/globals/index.xml

Settings

Start of /roxen/2.1/administrator/globals/settings.xml

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.

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.

A big disadvantage is that running Roxen threaded will make it vulnerable to all bugs in the operating systems thread implementation. This may make Roxen perform badly, hang Roxen or even hang the computer itself.

To enable threads Roxen must be started with `start --DENABLE_THREADS`, something that is done automatically on Solaris. 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.

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 that 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.

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.

End of /roxen/2.1/administrator/globals/settings.xml

ABS

Start of /roxen/2.1/administrator/globals/abs.xml



Globals / ABS

The ABS, or Anti Break 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.

End of `/roxen/2.1/administrator/globals/abs.xml`

Auto Restart

Start of `/roxen/2.1/administrator/globals/auto-restart.xml`



Globals / Auto Restart

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.

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

End of `/roxen/2.1/administrator/globals/auto-restart.xml`

Cache

Start of `/roxen/2.1/administrator/globals/cache.xml`



Globals / Cache

The *Global / 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 num-

ber 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.

End of `/roxen/2.1/administrator/globals/cache.xml`

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 Pike or Java 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.

End of /roxen/2.1/administrator/modules/index.xml

Installing

Start of /roxen/2.1/administrator/modules/installing.xml

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.

End of /roxen/2.1/administrator/modules/installing.xml

Module Types

Start of /roxen/2.1/administrator/modules/types.xml

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.

End of /roxen/2.1/administrator/modules/types.xml

File Systems

Start of /roxen/2.1/administrator/modules/filesystems.xml

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.

End of /roxen/2.1/administrator/modules/filesystems.xml