



*Urchin™ ISP 1.3 for Irix
Installation and Administration Guide*

QUANTIFIED
WEB SYSTEMS, INC.

2165 India Street
San Diego, CA 92101 USA
619 233-1400 fax 619 233-6510

Product: Urchin ISP
Version: 1.3-Irix 6.2 May, 1998

Copyright 1998 Quantified Web Systems Inc., 2165 India Street, San Diego, CA 92101 USA. All rights reserved.

This product and document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by and means without prior written authorization of Quantified and its licensors, if any.

Quantified, Quantified Web Systems, Inc., Quantified Systems, Urchin, Urchin ISP, the Quantified logo, and the Urchin logo are trademarks, registered trademarks, or service marks of Quantified Systems, Inc. SGI, Irix, and the SGI logo are trademarks, registered trademarks, or service marks of Silicon Graphics, Inc. The Netscape Communications Corporation logo, Netscape, and Netscape Navigator are trademarks of Netscape Communications Corporation. All other trademarks are property of their respective holders.

DOCUMENTATION AND SOFTWARE ARE PROVIDED "AS IS." QUANTIFIED'S LICENSOR, QUANTIFIED SYSTEMS, INC. ("QUANTIFIED"), MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THE SOFTWARE. QUANTIFIED DOES NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATIONS REGARDING THE USE OR THE RESULTS OF THE USE OF THE SOFTWARE IN TERMS OF ITS CORRECTNESS, ACCURACY, RELIABILITY, CURRENTNESS, OR OTHERWISE. THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE IS ASSUMED BY YOU. IN NO EVENT WILL QUANTIFIED, AND ITS DIRECTORS, OFFICERS, EMPLOYEES, OR AGENTS, COLLECTIVELY QUANTIFIED, BE LIABLE TO YOU FOR ANY CONSEQUENTIAL, INCIDENTAL, OR INDIRECT DAMAGES INCLUDING DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, AND THE LIKE, ARISING OUT OF THE USE OR INABILITY TO USE THE SOFTWARE EVEN IF QUANTIFIED HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. PLEASE DON'T SUE US, IT ISN'T NICE.

Contents

Preface	v
1. Urchin ISP for Irix Overview	1-1
What's New with 1.3	1-1
Operational Overview	1-2
Irix Interaction	1-3
Webserver Interaction	1-3
2. Installation and Configuration	2-1
System Requirements	2-1
Four Step Installation	2-2
Run Urchin and Install License.	2-5
3. Log Formats and Webrowsers	3-1
What Formats Does Urchin Use?	3-1
Configure Netscape for Extended Combined Log Format	3-2
Configure Apache for Extended Combined Log Format	3-2
Configure NCSA for Extended Combined Log Format	3-3

4. The "How To" Section	4-1
How To Add a New Site	4-1
How To Edit or Delete a Site	4-2
How To Filter: Create Sub-Reports, Exclude and Include	4-2
How To Export Data to a Database	4-3
How To Upgrade from Urchin 1.2.x	4-4
5. Configuration Directives	5-1
Global Configuration Directives	5-1
Virtual Host Directives	5-2

Preface

The *Urchin ISP 1.3 for Irix Installation and Administration Guide* contains instructions and information to install, upgrade, and configure Urchin ISP 1.3 software on the Irix platform.

This guide is intended for experienced Irix webmasters and system administrators who:

- Are familiar with the Irix operating system.
- Understand the webserver configuration on the system.
- Have superuser (`root`) access on the system.

How This Guide Is Organized

This guide is organized into five chapters:

Chapter 1, "Urchin ISP for Irix Overview," is a general explanation of the features and functions of Urchin and its day-to-day operation and interaction with the webserver and operating system.

Chapter 2, "Installation and Configuration," contains step-by-step instructions for installation of the Urchin ISP 1.3 software onto a Irix system.

Chapter 3, "Log Formats and Webservers," contains information on what

formats Urchin uses for log files and how to configure various webrowsers including Netscape, Apache, and NCSA.

Chapter 4, "The 'How To' Section" contains step-by-step instructions for various administrative tasks including adding and deleting sites.

Chapter 5, "Configuration Directives," contains a complete list of all the configuration options for both global settings and site specific directives.

Typographic Conventions

The following table describes the typographic conventions used in this guide.

Typeface	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output.	Edit the <code>config</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output.	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized Command-line variable; replace with a real name or value.	See the <i>User's Guide</i> for... These are <i>global</i> options. You <i>must</i> be root . Type <i>rm filename</i> .

Welcome to Urchin ISP

We would like to welcome you to Urchin ISP 1.3. If you have any questions or comments regarding the product or documentation, please e-mail to `urchin@quantified.com`. Thank you – the Urchin Guys.

Urchin ISP for Irix Overview



The version 1.3 release of the Urchin ISP is a software product targeted to Internet Service Providers and businesses that serve multiple websites on a server. Although not mandatory, Urchin ISP is designed to run directly on the webserver. By processing log files generated from the webserver, Urchin creates easy-to-read reports that your user community will find immensely valuable and useful.

Urchin uses either the Common Log Format or the Extended Combined Log Format as described in Chapter 3, and is designed to work with most webservers including Netscape Enterprise and FastTrack, Apache, and NCSA.

By running each night, Urchin keeps log files small, reports current, and minimizes system loading. Designed to be completely automated and self cleaning, once configured, Urchin requires virtually no maintenance.

What's New with Urchin 1.3?

Urchin ISP 1.3 includes these new features:

- Improved speed and performance
- Extended Combined Log Format support
- Filtering and subreport include/exclude capability

- Improved Browser and Platform reporting
- Ability to easily export data to a database
- Flexible restart scripts
- Visitors statistics
- Directory statistics
- Complete listings on all data types

These new features combined with a powerful new stat processing engine make Urchin ISP 1.3 the optimal package for Service Providers, busy websites, or anyone that wants to get great stats from their site(s).

Since Urchin's reports are HTML based, users can view their reports via the web. They are reminded to do so by the weekly e-mailed Report Summary, which provides a synopsis of site activity and a link to the complete on-line report. There is no need for you to continually alert your users to Urchin's presence; they are notified automatically as Urchin runs. This benefits the ISP in many ways. Clients see the results of their work, and are inspired to maintain their sites, and, of course, pay their hosting fees. ISPs that bill according to bandwidth have a simple way to check usage. Simply install, configure and enjoy the benefits. Maybe have a drink or a cigar..

Operational Overview

Once installed, Urchin runs each night, processing log files created by your webserver. The data from the log file is extracted and entered into the weekly, monthly, and annual reports. After processing, log files are automatically deleted (unless the `save_logs` directive is used), keeping hard drive usage in check. The following is a more detailed account of a typical daily operation.

Urchin is started automatically each night from the `cron` function on your system. Urchin then reads the configuration file, `config`, and begins processing the sites one at a time. There is no limit to how many sites can be handled. The log files of each site are processed into the monthly, weekly, and annual statistics. New reports are generated and the log files are removed. Once all the sites are taken care of, Urchin ISP creates a special Webmaster Report that gives a summary of complete server activity. The

webserver is then restarted so it can re-open the log files.

Urchin automatically determines if it's time to send weekly e-mailers. Also, for each new site in the configuration, Urchin will automatically create the icons (images) and directories it needs to build the report.

Irix Integration

Urchin uses certain built-in functions of the Irix operating system. It is important to be aware of which functions these are in order to ensure consistent operation. The following is a list of functions that Urchin will use:

cron daemon	/usr/sbin/cron is usually running
mail program	/usr/bin/mail or equivalent
shell script	/bin/sh or equivalent

All of these functions are standard issue with Irix. If there is a problem running Urchin, you might check that these are available or running for the user id that runs Urchin.

Webserver Integration

There are two issues to consider when installing Urchin on a particular webserver: the log file formats and restarting the server. Urchin uses either the Common Log Format or the Extended Combined Log Format as described in Chapter 3, and is designed to work with most webrowsers including Netscape Enterprise and FastTrack, Apache, and NCSA. Consult our website, **www.quantified.com/urchin**, for information. Check to see which log format your server is generating. There are instructions in Chapter 3 on how to configure your webserver to generate the Extended Combined Log Format so that you can capture referral, platform, and browser data.

Most major webrowsers have a lock on the log files. This requires that the webserver is restarted or signaled to reopen the log files after Urchin processes the logs. We have provided a shell script with Urchin to accomplish this function. There are instructions and samples on configuring the shell script. Urchin will automatically call this script after it is done processing the logs.



To ensure a successful installation, please go through all the steps outlined in the three sections of this Chapter. The first section describes the requirements of server hardware and software. This is followed by the four-step installation process. After installation, we recommend that you test run Urchin as described in the third section of this Chapter.

Urchin 1.3 is not backward-compatible with Urchin 1.2.3 or before. If you are upgrading from a previous release, please see the "How To Upgrade From 1.2.x" section in Chapter 4.

System Requirements

Urchin ISP 1.3 is a very efficient program and requires minimal disk space and memory. The distribution will use 2 MB of disk space. Each site's Urchin report directory will use anywhere from 1-8MB of disk space depending on how busy a site is. Urchin is completely self cleaning, and will never consume large amounts of disk space or create a runaway number of files.

While running, Urchin ISP will use approximately 2 MB RAM. Extremely busy servers may use extra memory to handle the large amount of data generated, but typically disk space and memory are not issues with most modern web servers.

Four Step Installation

The following four-step installation process can be performed from a client or directly on the webserver. Use a command tool or telnet to access the server.

Since Urchin will need to delete log files and restart the webserver, it is recommended to install and run Urchin as root. If you are not installing Urchin as root, you will need to check permissions on executables and logfiles. The following example assumes that Urchin is installed as root in the `/opt` directory.

Step 1: Download and Expand Archive

If you haven't already downloaded the Urchin archive, you should do so at this time from either a web browser or ftp at the following addresses:

```
http://www.quantified.com/urchin/download  
ftp://ftp.quantified.com/pub/urchin_isp/
```

Save the archive in the directory you have chosen for installation (i.e. `/opt`). You may need to be `root` to do this. Once you have downloaded the archive, follow the instructions below to expand the archive. The archive will automatically create an `urchin1.3` directory. To expand the archive, type the bold commands in a command tool or telnet window:

```
% su  
password: password  
# cd /opt  
# tar -xvf urchin_isp1.3_solaris2.5.tar
```

Step 2: Edit the Config File

The next step is to edit the configuration file to match your system setup. Make a list of the sites and location of log files for your server. In this

example `vi` is used to edit the configuration file. You can use another editor if desired. To edit the configuration file using `vi`, type:

```
# cd urchin1.3
# vi config
```

Edit the global information to reflect your setup. For a complete description of each Directive, refer to Chapter 5. You will need to decide on a directory for Urchin to create the Webmaster Report. For now, leave the License Directive the way it is. Set your global information by editing the bold text below:

```
Website_Provider:      Your_Company
WebMaster_Directory:  /webmaster_report_directory/
Mail_Program:         /usr/bin/mail
License:              LicenseStringGoesHere
```

For each website on your server you will need a Virtual Host set of Directives. Each site should have an entry that looks similar to the example below. Be sure to enter the actual directories and names for your sites. For a complete description of the Directives, refer to Chapter 5.

```
<VirtualHost>
Website_URL:          http://www.site1.com
Report_URL:           http://www.site1.com/logs/
Report_Directory:     /www/site1/logs/
Access_Log:           /www/site1/logs/access-log
Error_Log:            /www/site1/logs/error-log
Customer_Name:        Site1_Company
Customer_Email:       Joe@site1.com
</VirtualHost>
```

Write and quit the `config` file. To change configuration information later, simply re-edit the `config` file.

Step 3: Edit the Restart Script

The next step is to edit the restart script in the same directory. The restart script is called by Urchin after all the log files have been re-located for processing. This script is designed to either restart the webserver or signal the webserver to reopen the log files. Samples have been provided for various servers. Simply uncomment the lines that apply to your webserver (by removing `#`s) and edit the paths to match your webserver location.

```
# vi restart.sh
```

For example, if you are using Apache, uncomment the Apache line and edit the command as follows:

```
#####
#* For apache servers edit and uncomment...

kill -HUP `/usr/bin/cat /opt/apache/logs/httpd.pid`;

#####
```

If you are not sure what the restart command is for your server, test run the command from the command line and check that the server has restarted.

Step 4: Make Entry in Crontab

The last step in the installation process is to make an entry in `cron` so that Urchin can run each night. Use the following commands to enter Urchin into `cron`:

```
# setenv EDITOR vi
# crontab -e
```

Add the following line to the end of your crontab replacing /opt with the actual path. (Those are zeroes at the beginning of the line.)

```
0 0 * * * /opt/urchin1.3/urchin > /opt/urchin1.3/log 2>&1
```

If you are upgrading be sure to remove previous Urchin entries. Write and quit the file.

Run Urchin and Install License

Although Urchin will run automatically each night, it is advisable to test run Urchin from the command line to ensure proper operation. The following procedure will accomplish both testing and licensing. The first step is to run Urchin from the command line. Please note that if you have large log files that have accumulated over a long period of time, it may take a while for Urchin to run this first time.

```
# sh
$ /opt/urchin1.3/urchin
```

When Urchin is done processing the log files, you will see either a successful completion statement or an error message. Most error messages will result from a misconfiguration or permissions problem. If there is an error, such as a missing directory or invalid permissions, please make the correction before continuing.

After successful completion, check to make sure the webserver has restarted properly.

Licensing

To license your copy of Urchin, view one of the newly produced Urchin reports with a browser such as Netscape. Use one of the `Report_URLs` from your `config` file as the location. At the top of the report, you will see a link, `[Click here to license]` Simply click on the link and the browser will take you to our on-line registration center.

Follow the on-line instructions to complete the licensing process. Your license will be e-mailed to you automatically. Once you receive the license, copy and paste the license string from the e-mail message to the config file.

```
# vi config
```

In the `config` file, locate the `License` directive and replace the text "LicenseStringGoesHere" with your license string similar to:

```
Website_Provider:      Your_Company
WebMaster_Directory:   /webmaster_report_directory/
Mail_Program:          /usr/bin/mail
License:               AZAWOIJ2AZIWEPB3FSW90AFZMMA
```

That's it, the rebels are there! The `[Click here to license]` link will be gone the next time Urchin runs. You can either run Urchin from the command line or wait until Urchin runs that night.

Note: If you do not license Urchin, it will expire in 60 days.

This section describes the formats of acceptable logs files for Urchin and how to configure your webserver to get the most out of Urchin. It is possible to use Urchin with other servers such as Real Audio™ and proxy servers as long as the log configuration is consistent with the formats outlined below.

What Formats Does Urchin Use?

Urchin 1.3 is compatible with both the Common Log Format and the Extended Combined Log Format as described by NCSA. One of these will usually be the default; if not, most servers can be easily configured to create these formats.

The Common Log Format is a single line format that contains information on what domain or IP address is making the request, when the request was made, what was requested, and the size/status of the request. The Extended Combined Log Format is simply an extension of the Common Log Format. In addition to the above information, the Extended Combined Log Format logs information on where the requester came from and what hardware/software he or she was using.

Therefore, in order to capture the maximum data for your reports, it is desirable to configure your server to use the Extended Combined Log

Format. The following sections outline how to configure various popular servers to use the longer format. If your server is not listed below, you may find that you can apply one of these procedures to your server.

Note: It is possible to run Urchin on log files that contain either too few or too many fields of information. Urchin will ignore missing or extra data.

Configure Netscape Enterprise/FastTrack for Extended Combined Log Format

To change the log format to the Extended Combined Log Format, use the Administration Server via a browser to access the configuration. For each server on the machine which is listed below the "Servers Supporting General Administration" banner, follow these steps:

1. Click on the button for a particular server.
2. Click on the "Server Status" button at the top.
3. Click on the Log Preferences link on the left.
4. Scroll in the middle until you find the Format: options.
5. Select the Only Log option. Note: you may need to stop the server or change to the name of the Log File for this step.
6. Check the first 8 boxes directly below the Only Log option up to and including HTTP header, user-agent.
7. Click the "OK" button at the bottom.
8. Click the "Save and Apply" button at the bottom.

Repeat the above steps for each server.

Configure Apache for Extended Combined Log Format

To change the log format to the Extended Combined Log Format, you will need to edit the `httpd.conf` file in the Apache configuration. This is located in the `config` directory within the Apache distribution which is typically installed in `/usr/local/`, `/usr/local/etc/`, or `/opt`.

Once you have located the file, you will need to add the following line to the `httpd.conf` file. Add the line *before* any `<VirtualHost>` entries. This

will set the format globally for all sites. Add the line exactly as it appears below including quotations. Note: in the upcoming version 1.3 of Apache, you may be able to use "Combined" instead of the ridiculous string.

```
LogFormat "%h %l %u %t \"%r\" %s %b \"%{Referer}i\" \"%{User-agent}i\""
```

If you were using the `AgentLog` and `RefererLog` directives in the configuration, you may remove these as they are no longer necessary. After making this change, you will need to restart the server. You can use the following line to restart Apache. Replace the `/opt/apache/logs` with the actual location of the PID file.

```
kill -HUP `cat /opt/apache/logs/httpd.pid`
```

Configure NCSA for Extended Combined Log Format

To change the log format to the Extended Combined Log Format, you will need to edit the `httpd.conf` file in the NCSA configuration. This is located in the `conf` directory within the NCSA distribution which is typically installed in `/usr/local/`, `/usr/local/etc/`, or `/opt`.

Once you have located the file, you will need to add the following line to the `httpd.conf` file. Add the line *before* any `<VirtualHost>` entries. This will set the format globally for all sites.

```
LogOptions Combined
```

After making this change, you will need to restart the server. You can use the following line to restart NCSA. Replace the `/opt/httpd/logs` with the actual location of the PID file.

```
kill -HUP `cat /opt/httpd/logs/httpd.pid`
```

This section covers the basic things you may need to do during administration of Urchin. After making changes to the Urchin configuration, you do not need to restart Urchin. Urchin runs automatically from the `cron` daemon and will read in the configuration each time.

How to Add a New Site

Adding sites to the Urchin configuration is easy and straightforward. The first step is to add the site to your webserver operation. Make a note of the location of the log files. Then add the following lines to the Urchin `config` file replacing the paths with your specific information. For a complete description of each directive, please refer to Chapter 5.

```
<VirtualHost>
Website_URL:           http://www.newsite.com
Report_URL:            http://www.newsite.com/logs/
Report_directory:     /www/newsite/logs/
access_log:            /www/newsite/logs/access-log
error_log:             /www/newsite/logs/error-log
Customer_Name:         New_Site_Inc
Customer_Email:        Joe@newsite.com
</VirtualHost>
```

How to Edit or Delete Sites

You can make changes to the Urchin configuration at any time. Simply edit the `config` file and your changes will be loaded the next time Urchin runs.

To delete a site from the Urchin configuration, simply remove the `VirtualHost` entry for that site starting with `<VirtualHost>` and ending with `</VirtualHost>`. Your changes will be recognized the next time Urchin runs.

WARNING: Make sure that your webserver configuration is accurate. If you are deleting a site from your webserver, be sure to delete the site from the webserver configuration and not just the content and log files. A problem can occur if you remove the log directory for a particular site but do not remove the site from the webserver configuration. The next time Urchin runs, it will attempt to restart your webserver. This operation can fail if the log files cannot be opened because the directory doesn't exist.

How To Filter: Create Sub-Reports, Exclude and Include

It is perfectly acceptable to have different `VirtualHost` entries use the same log files. Urchin will recognize this fact and reuse the log files for each occurrence. This allows us to create multiple reports from the same log file. Using filters, these reports can be tailored to a specific need. For each report you will need a separate `VirtualHost` entry and a separate `Report_Directory`.

For example, let's say you have your main site, `http://www.serious.com` located in the directory `/www/serious/`, and you have a subsite for a friend in a directory off your site called `smurfs`. Now this is a little embarrassing when you are trying to show the boss how much traffic `serious.com` is doing and there is an entry for `smurfs`. So, you want to create separate reports for the boss and your friend using the same log file. To do this, you would make entries in the Urchin `config` file similar to:

```
#main serious report
<VirtualHost>
Website_URL:           http://www.serious.com
Report_URL:            http://www.serious.com/logs/
Report_directory:      /www/serious/logs/
access_log:            /opt/apache/logs/access-log
error_log:             /opt/apache/logs/error-log
Customer_Name:         Serious Business Inc.
Customer_Email:        me@serious.com
Filter_Out:            smurfs
</VirtualHost>

#smurfs report
<VirtualHost>
Website_URL:           http://www.serious.com/smurfs
Report_URL:            http://www.serious.com/smurfs/logs/
Report_directory:      /www/serious/smurfs/logs/
access_log:            /opt/apache/logs/access-log
error_log:             /opt/apache/logs/error-log
Customer_Name:         Smurf Museum
Customer_Email:        friend@serious.com
Filter_In:             smurfs
</VirtualHost>
```

Since each hit for your friend's site will contain "smurfs" in the path, we can use the `Filter_Out` directive to *exclude all entries that contain the name*; and use the `Filter_In` directive to *only include entries that contain the name*. For more information on filtering, see the next Chapter for each directive.

How to Export Data to a Database

Urchin automatically creates sorted data ready for importing into your favorite database. The format is tab-separated with the first line naming the fields. This can be immediately imported into most databases including Filemaker Pro, Access, etc.

To export data from Urchin, use a browser to view the report. At the bottom of each section there is a link, "View Complete Data." This link will take you to the exportable data. You can use the "Save as..." option with your browser to save the file to your computer. On a Windows machine, you can right-click the link and save the data without viewing it. Once the data is saved on your machine, you should be able to open or import the data from your database application.

With the reports being monthly in nature, to build a complete and non-duplicating dataset, wait until a month's reporting is finished before importing it into your database. Urchin keeps the monthly reports in place for one year.

How to Upgrade From 1.2.x

Out of necessity, Urchin1.3 is not backward compatible with 1.2 versions. In order to avoid data loss, it is recommended to create a new urchin report directory and switch between using 1.2 to 1.3 on the second day of the month after Urchin 1.2.x has finished with the previous month.

Go through the installation process as described in Chapter 2. Be aware that there are subtle changes in the `config` file, especially regarding `access_log` and `referer_log`. The character between "access" and "log" and "referer" and "log" **must** be an underscore, not a hyphen. **Previous versions of Urchin used a hyphen. This has been changed for consistency reasons.** It is recommended to create a new `config` file and not reuse one from a previous version.

Be sure to remove the previous Urchin from `crontab` when you enter the new one.

The entire Urchin configuration process is handled by the `config` file located in the Urchin distribution. This file contains configuration directives that tell Urchin how to run and what to process. There are two types of configuration directives: global and virtual host directives. The global directives guide the entire operation of Urchin including the Webmaster Report, whereas the virtual host directives are site specific. They define the location, URL, and other information associated with a specific site.

Global Configuration Directives

The following global configuration directives should appear once and only once in the `config` file. Each directive should be on its own line with the corresponding information following a colon, ":".

◆ **Website_Provider**

This directive can be set to the Company or Department that is responsible for the web servers. The clients will see that their reports came from this source. You will need to use an underscore instead of whitespace in the name. Example:

```
Website_Provider:  Foo_Internet_Services,_Inc.
```

◆ Webmaster_Directory

This directive specifies where to create Urchin's unique Webmaster Report, which gives a summary of all www activity on the server. This report tracks bandwidth and is particularly useful for billing and comparison purposes. Urchin will create files and subdirectories in this directory including `index.html`, `icons`, etc. To view the report, point your browser at this directory. Be sure to use the complete path to the directory. Example:

```
Webmaster_Directory:  /www/webmaster/
```

◆ Mail_Program

This directive indicates where Urchin can find a mail program so that it can send e-mail to the customers each week. This is typically `/usr/bin/mail`, and should not need any modification. If you have trouble sending e-mail, try testing the mail program by typing:

```
/usr/bin/mail you@youraddress.com < somesmallfile
```

You will most likely not need to edit this directive. Example:

```
Mail_Program:  /usr/bin/mail
```

◆ License

This directive contains the license code for your copy of Urchin. The license code is specific for each machine. To obtain a license, see the end of Chapter 2. When you receive your license via e-mail, replace the temporary license code with your license code. Example:

```
License:  AADFLKJAYOURCODEIVXOAWI
```

Virtual Host Directives

The virtual host directives are site specific parameters. For each site on your server that Urchin will create a report for, you will need a set of virtual host directives within the `config` file similar to:

```
<VirtualHost>  
    directive  
    directive  
    directive  
    ....  
</VirtualHost>
```

Each virtual host entry will need to have a unique `Report_Directory`. Other directives are optional and do not necessarily need to be unique.

◆ **Website_URL**

This directive specifies the website for which a given report is being generated. The client will see this directive at the top of the report, and on the weekly e-mail update. Example:

```
Website_URL:  http://www.client1.com
```

◆ **Report_URL**

This directive specifies where the client can find his or her report on-line. Because Urchin creates HTML based reports, you will want to have the client's report in a place where they can access it via the web. For example, if the client's website is in the `/www/client` directory, you might choose to create a subdirectory within the client directory for the report.

If necessary, reports can be password protected using the password protection configuration of your webserver, such as `.htaccess`, used by Apache. Example:

```
Report_URL:  http://www.client1.com/urchin
```

◆ Report_Directory

Directly related to the `Report_URL`, this directive specifies the physical directory where Urchin will create the report. This parameter must be unique for each site in the Urchin configuration. Urchin will create files and subdirectories within this directory (`index.html`, `icons`, `html`, `txt`, etc.). Although not necessary, it is acceptable to have the log files in the same directory as the Urchin report. Example:

```
Report_Directory:  /www/client1/urchin/
```

◆ Access_Log

This directive specifies the physical location of the log file to use for this report. This log file is generated by the webserver and should be either in the Common Log Format or the Extended Combined Log Format as described in Chapter 3. Urchin will re-locate this file, process it, remove it, and restart the webserver. If two virtual host entries use the same log file, Urchin process both reports before deleting the file. Example:

```
Access_Log:  /www/client1/logs/access
```

◆ Error_Log

Besides the normal access log file, Urchin will also process an error log file if desired. Simply point this to the error log file created by your server. Example:

```
Error_Log:  /www/client1/logs/errors
```

◆ Customer_Name

Set this directive to the name of the company or individual for whom the report is being created. This directive is used by the Webmaster Report. In this way, a copy of the Webmaster Report can be used without knowing the URLs of each client. Use an underscore instead of whitespace. Example:

```
Customer_Name:  Foo_Bar_Inc.
```

◆ Customer_Email

Set this directive to a valid e-mail address for the customer. Urchin will automatically e-mail a summary and URL of the report to this person.

```
Customer_Email: joe@foobar.com
```

◆ Filter_In

This directive can be used to create subreports or to simply filter which hits from the server are reported on. This directive is optional, and if set specifies a case-insensitive filter for each log line that is processed. Only log entries that contain the filter will be reported on. Please see the sample in Chapter 4 on filtering and subreports. Typically, this operation is used to create a report on all the hits that occurred for a particular directory within a site. For example, if you wanted to create a report that contained information only on a directory called `foo`, you would enter the following into the report's configuration entry:

```
Filter_In: foo
```

◆ Filter_Out

The opposite of the previous directive is the `Filter_Out` directive. Again, this directive is optional and can be used to create a subreport or simply omit certain data from the report. The filter is case insensitive. Any log entry that contains this filter will be omitted from the report. Please see the sample in Chapter 4 on filtering and subreports. This filter can be used to filter out entire directories, individual files, or even a particular host. You can only use this filter once per virtual host. If you need to filter two separate pages, either place them in a special directory that you can filter or have them named similarly. Example:

```
Filter_Out: blank.html
```

◆ Save_Logs

This directive is optional and can be used to stop Urchin from deleting the log files. If this directive is set to "Yes", then Urchin will move the logfile by adding the extension, ".urchin.saved", to the name. If you wish to run a different package on the log file (we can't imagine why you would), you can do so with this directive. Point the secondary package at the ".urchin.saved" file. Be aware that Urchin will overwrite the ".urchin.saved" file each night, so the secondary package will need to run daily as well.

Data can be imported directly from the Urchin reports into a database for further mining in which case you would not need to use this directive. But if you must run another stats package, simply use:

```
Save_Logs:  Yes
```


