

Kensho Guide to Site Administration

Copyright © 1995, 1996 Sean Dreilinger
Version 1.12

This document contains proprietary information that is commercially sensitive. It is strictly for issue to selected parties only, and is confidential to them and their professional advisers. Acceptance of a copy of this document signifies assent to the following undertakings:

1. No person should disclose any information contained in this document to any other party, except in the course of seeking professional advice in connection with it. The same undertakings as to confidentiality should be required of any prospective professional adviser before any particular is disclosed to such a person.
2. The document, having been given to a specific individual, is not to be handed on to any other person except in the course of seeking professional advice.
3. If the document is no longer needed by the person to whom it was given, it must be returned promptly.
4. This document is copyright and must not under any circumstances be copied.

Any person coming into possession of a copy of this document who does not agree to the foregoing conditions, should return the document forthwith to the issuer.

Contents

1	Introduction	1
1.1	Purpose	3
1.1.1	Continually Evolving Medium	3
1.1.2	Support Calls	4
1.1.3	Software and Information Resource	5
1.1.4	Strange Formatting?	5
1.2	Before This Makes Any Sense...	7
1.2.1	UNIX	7
1.2.2	HTML	7
1.2.3	Networking	8
2	Design Overview	9
2.1	General Site Concepts	11
2.1.1	User-Space Organization	11

2.1.2	Administrative–Space Organization	16
2.1.3	Room To Grow	19
2.2	Site—Specific Information For <code>www.client–domain.org</code> .	21
2.2.1	Client Project History	21
2.2.2	Kensho Support	21
2.2.3	ISP Information	22
2.2.4	Software and Hardware Specifics	24
2.2.5	Laptop–Ready Site Archive	25
2.2.6	Statistics Online	26
2.2.7	Information Service Frameworks	26
3	Tasks—How To	33
3.1	Connect To Your Website for Remote Administration . .	35
3.1.1	<i>telnet</i> and <i>rlogin</i>	35
3.1.2	Client Software	35
3.2	UNIX Housekeeping	41
3.2.1	<i>rm -rf, *</i> , and Other Scary Stuff	41
3.2.2	Tildes	41
3.2.3	Filename Completion	42
3.2.4	<i>motd</i> —Daily News, ISP–style	42
3.2.5	E–mail	42
3.2.6	Free Space	43

3.2.7	ISP Newsgroups	45
3.2.8	Shell Configuration	46
3.2.9	More Information	48
3.3	Change an Existing HTML Page	49
3.3.1	Revision Control System Checkout	49
3.3.2	Edit in <i>joe</i>	49
3.3.3	<i>weblint</i>	51
3.3.4	Test In Browser	51
3.3.5	<i>rcs</i> Checkin	52
3.3.6	More Information	52
3.4	Transfer Files To And From Your Site	53
3.4.1	<i>ftp</i> Software Options	53
3.4.2	Compression Software Considerations	56
3.4.3	More Information	62
3.5	Change The ‘Automatic Overnight’ Stuff	63
3.5.1	The <i>crontab</i> Command	63
3.5.2	Suggestion: Call One Script	64
3.5.3	Testing Before You Go Unattended	64
3.5.4	More Information	65
3.6	Re–Build the Searchable, Fulltext Website Index	67
3.6.1	Using <i>swish</i>	67

3.6.2	More Information	68
3.7	Create An Electronic Mail Discussion List	69
3.7.1	Questions to Ask Yourself	69
3.7.2	<i>createlist</i>	70
3.7.3	<i>rc.sitename.smartlist</i>	72
3.7.4	<i>rc.custom</i>	73
3.7.5	<i>subscribe.txt</i>	74
3.7.6	<i>x-command</i>	75
3.7.7	Test the List	76
3.7.8	Set up Hypermil Index	76
3.7.9	Register Public Lists With PAML	77
3.7.10	Removing a List	77
3.7.11	More Information	78
3.8	Working With Virtual E-mail Addresses	81
3.8.1	What Is A Virtual E-mail Domain?	81
3.8.2	Limitations Of Virtual E-mail Domains	82
3.8.3	Configuring User Mail Agents	83
3.8.4	Cooking With <i>procmil</i> : Recipes	84
3.8.5	Test the Address(es)	87
3.8.6	Revision Control	88
3.8.7	More Information	88

3.9	Create A Password–Protected Area	93
3.9.1	Permissions Caution!	93
3.9.2	Procedural Overview	93
3.9.3	Create/Edit The Access File	93
3.9.4	Create Users and Password(s)	96
3.9.5	Test the Restrictions	97
3.10	Getting Attention	99
3.10.1	Coming Up First In Full–text Indices	99
3.10.2	Registration and Listing	102
3.10.3	Notifying Existing Clients	104
3.10.4	Advertising	105
3.11	Adding or Changing Graphics	107
3.11.1	Software	107
3.11.2	Text	108
3.11.3	File Formats	109
3.11.4	More Information	110
4	Software	111
4.1	<i>W</i> ³ System Administration	113
4.1.1	<i>cyclelog</i>	113
4.1.2	<i>getgraph</i>	114
4.1.3	<i>getstats</i>	115

4.1.4	NCSA <i>httpd</i>	116
4.1.5	Netscape <i>httpd</i>	117
4.1.6	<i>swish</i>	117
4.1.7	<i>wwwwais</i>	119
4.2	W ³ Content Administration	121
4.2.1	<i>giftrans</i>	121
4.2.2	<i>weblint</i>	121
4.2.3	<i>rtfohtml</i> and <i>rtfoweb</i>	122
4.3	W ³ Access	125
4.3.1	<i>lynx</i>	125
4.3.2	NCSA <i>Mosaic</i>	126
4.3.3	Netscape Navigator	127
4.4	Internet Utilities	129
4.4.1	<i>telnet</i> and <i>rlogin</i>	129
4.4.2	<i>ftp</i>	129
4.5	Electronic Mail	131
4.5.1	<i>hypermail</i>	131
4.5.2	<i>procmail</i>	132
4.5.3	<i>smartlist</i>	133
4.6	General	135
4.6.1	GNU Utilities	135

4.7	Helpful Resources	139
4.7.1	HTML Standard	139
4.7.2	CGI Standard	140
4.7.3	Mailing List Management Software FAQ	141
4.7.4	W ³ Newsgroups	141
5	Endnotes	143
6	Index	145

Chapter 1

Introduction

Welcome! If you've been handed (or e-mailed, *ftp*'d, etc.) a copy of this manual, you're involved in improving and maintaining a collection of Internet services that were designed and built by Kensho, LLC.

1.1 Purpose

This document is intended to save you time in administering a Kensho-built site, and ultimately to save everyone time, because we'll all be able to work from the same consistent guideline.

1.1.1 Continually Evolving Medium

To finish a work? To finish a picture? What nonsense!
To finish it means to be through with it, to kill it, to rid it of its soul, to give it its final blow... the coup de grâce for the painter as well as for the picture.

Pablo Picasso (1881–1973)

As you read this, things are already changing, improving, going out of date, it is the nature of information, and consequently a fact of life in our business. Please absorb the *principles* and philosophy of this site's layout, design, and administration as you peruse this document, then double-check the online version before doing anything drastic to our client's beloved info-service. As the ultimate intelligent agent—a human—you may be the first to identify important updates and improvements for this document. Please share the wealth with other members of the support@kensho.com discussion list.

The up-to-the-minute version of this document is accessible via the world wide web at this URL: <http://www.kensho.com/support/vsag/>.

To view it, please use the access information from Table 1.1.

Table 1.1: Access to the Online Guide

USERNAME	PASSWORD	MNEMONIC
username	*****	memorable

1.1.2 Support Calls

By making printed and electronic copies of this document available to our information-literate clients, We hope to reduce the number of support calls we get—especially those that turn out to be frequently-answered technical questions or exchange of other information which could be more easily shared in this format.

For issues which you can't easily resolve, don't hesitate to call, write, fax, or otherwise contact us for assistance. The current client maintenance/support information appears in Table 1.1.2.

Table 1.2: Kensho Support Contacts

E-MAIL:	support@kensho.com
VOICE:	+1 (619) 514-3939
FAX:	+1 (619) 514-3938

On-site support visits are arranged through the above channels. We'll welcome walk-in appointments and host occasional classes when we move from our Balboa Avenue location to the Chesapeake facility.

1.1.3 Software and Information Resource

We've taken great care to identify and offer links to the authoritative Internet documentation & distribution points for much of the software you'll use on a daily basis. Whenever an in-depth explanation of an application or concept is required, have at it! After the obligatory RTFM period, feel free to contact the software authors directly and share your experiences with the support@kensho.com list.

1.1.4 Strange Formatting?

In certain versions of this document, you'll notice curious use of double underlining, red text, and outline boxes. This is to automatically generate hyper-links during conversion to HTML.

1.2 Before This Makes Any Sense...

Like many technical documents, this one is built upon the domain knowledge of several related fields. To cruise through this one with the greatest of ease, you should already be familiar with...

1.2.1 UNIX

Most of Kensho's Internet Services are constructed and run upon different flavors of the UNIX operating system. You don't need to be a UNIX system administrator, but it helps to be able to view the system through both user and administrative-level filters. A good general guide for working with UNIX at the user-level is *A Practical Guide to the UNIX System??*. A standard reference for UNIX system administration is Aleen Frisch's *Essential System Administration??*.

1.2.2 HTML

Platform independent front-ends to many Internet Services can be created using Hypertext Markup Language (HTML), and the related Common Gateway Interface (cgi) spec. The latest proposed HTML spec can be read and referenced via the W^3 Consortium at <http://www.w3.org/>.

1.2.3 Networking

To administer a single client's Internet Services, you may be required to connect and interact with a number of disparate services and information banks on machines that are physically located all over the world. To perform these tasks with full understanding and maximum comfort level, you need to have a strong grasp of computer networks and inter-networking. A good place to brush up on this is freely available *Big Dummy's Guide to the Internet* document published by the *Electronic Frontier Foundation* (EFF), on the web at <http://www.eff.org/>.

Chapter 2

Design Overview

Kensho attempts to build information services upon a stable foundation of simple rules for user interface (UI) design and information systems (IS) organization. In addition to general guidelines, this chapter will include any client-specific gotchas for the Internet site you'll be working on.

2.1 General Site Concepts

2.1.1 User–Space Organization

We’ll define ‘user–space’ as the part of our Internet Information Service that is actually seen and touched by end–users, whether they’re employees of the service provider poking around on an intranet, or public visitors exploring the free information offerings of a given organization. To help persuade the end–user to spend a little more time on the site, we’re going to incorporate simple concepts drawn from the academic fields of information–seeking behavior and user interface design.

Satisficing

People tend to quit while they’re ahead when seeking information. In other words, If they find a source of information that meets their need or interest, they will consider their quest fulfilled and stop looking. Our goal is to be discovered first, so they stop their search upon arriving at our information. Strategies that address the tendency of information seekers to *satisfice* are proposed in Section 3.10, the *Getting Attention* section of this manual.

Chunking

In keeping with the research of Miller??, we always attempt to organize and categorize the information we’re presenting to the end–user into seven or fewer categories, each of which may be subdivided in a

similar fashion. Short of an experienced waiter/waitress taking an order, people have great difficulty distinguishing between, remembering, and choosing from more than seven (\pm two) categories of information.

Berrypicking

From the work of Marcia Bates??, we've learned that information-seeking behavior (i.e. Internet Surfing) resembles berrypicking. Untrained information seekers look for information much the way people pick berries—a good one HERE, oh look! A good one THERE—and do not cover the field in any thorough fashion. The same applies to the typical pattern of information-seeking on the Internet. Users wander the Internet in search of attractive 'berries,' skipping from page to page, site to site. As administrators of that information, we help users to find our client's gold-nuggets of knowledge through *collocation*, *multiple formats* and *external indexing*.

Collocation Supporting collocation is easy. Wherever appropriate within an Internet site or service, add hyperlinks to related information that appears elsewhere on the site. This concept can be expanded to *linkbacks*—reciprocal links on other Information services.

Multiple Formats One person's *ripe* berry may taste *rotten* to another. Whenever possible, present the same information through each available sensory channel—visual, textual, auditory, etc.

Similarly, offer a variety of file formats whenever a valuable piece

of information is being presented—Postscript, PDF, word processor, etc. Each format will be of value to a particular segment of our target audience.

External Indexing Listing the information we're offering with as many Internet Directories and indices as possible is critical to drawing traffic and raising awareness. Hints to leverage directory and indexing services are offered in Section 3.10 of this manual.

Shneiderman's Rules

User Interface expert Ben Shneiderman offers some good ground rules, carefully developed before the explosive growth of the Internet??. These ground rules and principles apply nicely to our work.

Three Golden Rules of Hypertext

1. There is a large body of information organized in to numerous fragments
2. The fragments relate to one another
3. The user needs only a small fraction of the fragments at any time.

Five Factors for Evaluating User Interfaces

Time To Learn Can they learn it in six seconds? or do they need a seminar?

Speed of Performance McDonald's order-taking

Rate of Errors by Users Do they get it? obvious ones in log files

Retention Over Time

Subjective Satisfaction Did they enjoy it and will they return?

Hypertext

“Hypertext” is a concept coined in the 1960's by Ted Nelson. Research demonstrates that proper use of Hypertext media improves *retention* and *learning* beyond that attained when learning from traditional print media. Unfortunately, there is always the danger that careless use of hypertext will lead to hyper-chaos, rather than the desired benefits. Shneiderman forecasted this dilemma:

Poor design of hypertext is the more common problem; too many links are overwhelming, too many long articles makes reading dull and non interactive, long chains of links to reach required articles can be disorienting, and inadequate tables of contents or overviews make it difficult for users to determine what is contained in the Hypertext. Just because a text has been broken into fragments and linked does not insure that it will be effective or attractive. Successful hypertext, just like any successful writing project, depends upon good design of the contents. The hypertext author who creates a new work or the hypertext

editor who takes existing materials and puts them into hypertext form must take great care to produce an excellent document. Just as turning a theatre production into a movie requires new techniques of zooming, panning, close-ups, cuts, fades, etc., creating successful hypertext requires learning to use the features of the new medium.

Shneiderman *Designing the User Interface* p.411

Other Concepts

Many other traditional and common-sense design concepts apply well to the new media, here are a few more to keep in mind. . .

KISS Keep it simple.

Legible Avoid white text on black background, except for signage.

Culture Choose images carefully for an international audience.

Consider Physical Size Don't put your important action items, order buttons, etc.. at the bottom of the screen. Target a 640x480 pixel rectangle for your information, unless you have a special audience with big expensive computers—i.e. dedicated kiosks or art installation.

Visual Impact You've got about six seconds to get and keep an end-user's attention—the "back" button on a W^3 browser is just a click away. To understand the reality of this time limit, take a look at an access_log file and note that few browsers land on a single page

for more than about six seconds, then choose another URL on or off-site.

2.1.2 Administrative-Space Organization

Like the visible end-user face of the silicon system, the administrative side needs to stick to a few traditional and common sense guidelines.

Case Sensitivity

We've created directory structures for clients of both mixed and lower case. If you'll be working with the site using exclusively case-sensitive software, naming each directory on the client system with an Initial Capital letter will put directories first in file listings and help avoid accidental deletion of the directories.

If any case-insensitive software will be used in the site production or maintenance (Microsoft Windows 3.1, DOS, OS/2, Samba), then it is wiser to name files and directories in all lower case. Otherwise you may end up with a collection of browseable HTML files with case-sensitive URL's, sitting on top of a filesystem that has been transferred and translated between a few operating systems and is no longer wearing its original, carefully-mixed case. In other words—this can break all of your links.

Filename Length

This will be obvious to the users of MacOS and certain IBM OS's out there—you'll need to pay careful attention to the length of filenames, especially if all or part of the site will be maintained using Mac or IBM-based software. Several clients have asked that their entire W^3 site be transferable and presentable via DOS-based laptop computers. This requires careful planning and disciplined file naming!

UNIX aficionados might be content to name the 1994 Annual Business Report file like so:

```
1994_Annual_Business_Report.html
```

which will not translate elegantly to primitive operating systems. In fact, the name above is even one character too long to squeeze onto the MacOS HFS filesystem. Instead, consider adhering to the ancient 8.3 naming convention which works with the FAT filesystem found on the lowliest DOS and Windows 3.1 machines. Adhering to this, we'd name the 1994 Annual Business Report file:

```
94busrep.htm
```

Note that in order to design a website using *.htm* instead of *.html* as a default file extension may require modification of the web server's *resource configuration* file and *MIME types* file to accept the *.htm* extension as a MIME type of `text/html`.

Table 2.1 contains a comparison of filename lengths allowed by some popular operating systems.

Table 2.1: File Naming Considerations by OS

OS	FS	LENGTH	CASE RECALL?
OS/2	HPFS	255	N
DOS	FAT	11	N
WIN95	VFAT	255	Y
WINNT	NTFS	255	Y
UNIX	various	255	Y
MACOS	HFS	32	Y

Keep in mind that long filenames are hard to type accurately and can also cause grief in shell scripts or other applications where the filename length exceeds the maximum number of characters allowed in the `$PATH` or other environment variable.

Revision Control

As you'll see later on, Kensho applies some minimal *revision control* to the files it creates and maintains. This is a necessity when administration is shared among more than one worker, and can be a life-saver when the client calls with the disgruntled "We liked it better before, change it back!"

Using revision control, we can carefully track how our sites and services evolve, and when necessary, provide clients with detailed summaries (*ChangeLogs*) of the work we've performed. Of course, when a change is made that doesn't work or doesn't meet with the client's approval, putting things back in order is now a simple matter of restoring the old revision number, rather than

tracking down the person who did the editing and having them remember what needs to be put back.

Binary Formats

An executable program that runs on one of our sites will not necessarily work on another. We compile and maintain a large collection of software for our netsites, and this must be repeated for each operating system we work with. For example, the *getstats* program by Kevin Hughes, which analyzes the *access_log* file from a *W³* site, will only run on the platform it is compiled for. We can't move our *getstats* binary from a FreeBSD system to Linux, SunOS, IRIX, or Windows NT servers without recompiling the program from its source code.

For this reason, many of our virtual net sites have a *~/ftp/* or a *~/src/* directory that is used to keep compressed, customized source code. This comes in very handy when we migrate an entire site to another server or decide that additional customization is required.

2.1.3 Room To Grow

Some of the techniques and procedures described here will strike you as a bit much—too methodical, too anal-retentive, too careful, whatever. Please keep in mind that the long term goal is to always give the Information Service room to grow. You may move along to work with another site or project in a few months time, and we'd like the next person to be able to focus on growing the Information Service the same way you have, rather than wasting days trying to make sense of an undocumented 'one-of-a-kind' structure put in place by somebody now long-gone.

2.2 Site—Specific Information For www.client— domain.org

In this customized section we've explained the details for this particular net site—answering questions like “Who is the ISP for this site?” “Does this client have a Kensho Support arrangement?” “What code got hacked to put this puppy in working order, and where can I find that code?” etc.

2.2.1 Client Project History

Client—Executive—Contact initiated an Internet Presence project for Client in Date—Of—Start. Kensho—Salesperson and Kensho—Project—Manager of Kensho were brought in as consultants to design & develop the Internet services in co-operation with Client—Technical—Contact of Client. Kensho—Intern at Kensho was involved in prototyping the site in its first iteration. The site was made available on the Internet via password access during Date—Of—Test. Additional refinements were made under the direction of Client—Technical—Contact to meet Client's satisfaction. The site was made publicly available and registered with Internet Search resources and directories on Date—Of—Launch. Client—Technical—Contact will continue to improve and maintain the Client Internet presence.

2.2.2 Kensho Support

Support Contact

For non—urgent support queries, please send electronic mail to: support@kensho.com.

Terms

The proposal for Client specified that Kensho will provide support for a period of two months after completion of the Internet Presence.

2.2.3 ISP Information

Many Kensho clients are billed directly by an Internet service provider for some monthly recurring costs. This keeps costs down for our clients, but may present an added complication for those organizations without a clear understanding of the role ISP's play in the grand scheme of things—and why some ISP is sending them bills.

For the Client work, a mini-RFP describing the project was distributed to San Diego area ISP's. ISP-Name was selected by Client to provide Internet hosting and other services. Specific ISP information for the Client net site follows.

Contact Person

ISP-Contact-A at ISP-Name set up the UNIX accounts and DNS entries required for an InterNIC domain name registration. ISPCONTACTB at ISP-Name configured the initial Netscape Virtual W^3 server for www.CLIENTDOMAIN.
ISP-related questions should be sent to isp-email@isp-domain.net if at all possible.

ISP-Name

ISP-Address

ISP-City, ISP-State ISP-Zip

2.2. Site—Specific Information For `www.client-domain.org` 23

voice: ISP-Phone

fax: ISP-Fax

ISP-Name's Customer Service hours are 8:30am-7:30pm Monday-Friday, and 10:00am-2:00pm Saturday. You can also call the number above and report urgent problems during non-business hours and you will be called back within one hour.

Machines

All services and accounts for Client are currently housed on the machine `isp-domain.net`.

User Account(s), Groups, etc.

The following user accounts are available to Client on ISP-Name's server:

Table 2.2: ISP Access Information

USERNAME	PASSWORD	MNEMONIC
<code>username</code>	<code>*****</code>	<code>memorable</code>

In addition, the following e-mail forwarding aliases and lists are established:

All other e-mail arriving at the domain `client-domain.org` will remain in the `$DEFAULT` mailbox defined by *procmail*, which is currently the mailbox for the user `Client@isp-domain.net`. As more individual users & affiliates of

Table 2.3: E-mail aliases

info@client-domain.org	Client@isp-domain.net
postmaster@client-domain.org	Client@isp-domain.net
tlist@client-domain.org	<i>any and all recipients of this smartlist</i>
webstats@client-domain.org	<i>any and all recipients of this smartlist</i>

CLIENT get e-mail addresses with various ISP's such as CerfNet, MillenniaNet, Netcom, CTS, AOL, etc., the appropriate *procmail* aliases can be created via a simple process described later in this manual.

2.2.4 Software and Hardware Specifics

Server Platform and OS

The Client site is served from a 22MIPS Silicon Graphics Challenge server owned by ISP-Name. At the time of development, the host server was running SGI IRIX 5.3 operating system. The necessary W^3 binaries have been adapted, compiled and installed as necessary to work with this OS.

HTTP Server

ISP-Name operates a dedicated Netscape Communications virtual domain and web server for this site.

2.2.5 Laptop—Ready Site Archive

Every night, the latest version of the Client world wide web site is recursively compressed and saved into a PKZIP-compatible archive which can be downloaded from the Client ‘internal’ page at this URL: <http://www.client-domain.org/internal/>.

To uncompress the file, you’ll need a PKZIP-compatible unzip program, such as *INFO-zip* or *PKZIP* itself. Save the *inovaweb.zip* file in the location you’d like to use as the root of your ‘canned’ Client document tree and issue the following command:

```
pkunzip -d inovaweb.zip (if using PKUNZIP)  
unzip -dv inovaweb.zip (if using INFO-zip)
```

The file will expand to create a collection of subdirectories and files beginning with a folder named *www*.

After the unpacking is complete, you can start a web browser on the local computer and open the *home.htm* or *welcome.htm* file inside the *www* folder to begin browsing. The site search engine will not work on a canned installation, nor will the submission of online forms or any other links that require a live Internet connection to go off-site. If the laptop or local machine is attached to the Internet in some way, then the off-site links should also work, and the only part that will not function is the site-wide search engine. Other than that, you should have a good working copy of the www.client-domain.org site for presentations and meetings.

2.2.6 Statistics Online

The Client W^3 site compiles usage statistics (from raw hits) on a nightly basis, then automatically analyzes and charts the information. Client users can review the traffic data by browsing this URL: <http://www.client-domain.org/internal/stats/>.

On a monthly basis, a program such as *getstats* or *analog* is run to perform an in-depth analysis, which is then mailed to any and all recipients of the webstats@client-domain.org e-mail *smartlist*.

2.2.7 Information Service Frameworks

Several clever features of the www.client-domain.org site are partially complete, awaiting supplemental content and any additional design work desired by the Client team. In lieu of creating finished products laced with inaccurate or illegal information, Kensho has created several frameworks which can be used to aid in the implementation of these additional features if/when desired.

Customer Tutorial

A mockup customer tutorial is designed and installed in the User Education area of the site. Icons to create custom navigation bars are available, including dimmed and illuminated symbols for forward, back, 1,000x, 400x, and 100x magnification, negative view, and return to the menu. These should prove useful in the case that Client wants to achieve refined control over what users can do at each screen of the tutorial, and such constrained navigation may be a required ‘feature’ if academic accreditation is sought.

‘Welcome’ and ‘Menu’ pages for the tutorial are available and contain place-holder text. One example is online from beginning to end with place-

holding images and text. Additional examples can be created by duplicating the HTML source files and replacing the nonsense text and inappropriate images with well-written content and links to real-life images.

Feature Articles

Another idea for the site is to electronically publish original articles written by the Client principals and staff. While the exact format of the articles is undetermined (and may vary in media and implementation depending on the topic), a front-page location for promoting and linking the articles is designed into the Client Home Page. On the URL <http://www.client-domain.org/home.htm>, the upper, right-hand corner text is not part of the imagemap. Instead, this text can easily be changed to announce special offers, important news, or Client's feature articles.

Here is the relevant HTML source from the home.htm file, which can be quickly updated and linked to news or feature articles:

```
...
<TD ALIGN="RIGHT" VALIGN="middle"><H6><FONT SIZE=2><I>
Internet Services from the World Leader in...
</I></FONT></H6>
...
```

To create a front-page link to any feature article, special offer, or news item, simply edit the file home.htm and replace the bold text from our excerpt above with the hyperlink and descriptive text of choice. If it doesn't offend the aesthetic judgment or style guidelines at Client, this might be an appropriate place to use the NSHTML `blink` tag.

Abstracts and Reprints

Reprints of four academic papers are currently offered on the Clientuser-response form at this URL:

<http://www.client-domain.org/help/response.htm>

Checkboxes for users to request additional reprint request titles can easily be added by editing the source HTML of this form. Each fill-in item on the response.htm form is stored in a numbered cgi variable named *Qnn_descriptive_text*, where *nn* is the variable number beginning with 00 and *descriptive_text* is an easily understandable name for the variable.

Two relevant lines from the existing HTML response.htm form appear below:

```
...  
<BR><INPUT NAME="Q22_Paper_Hardarson_93" TYPE="CHECKBOX" ALIGN=left>  
<TEXTAREA NAME="Q23_Please_Read" ROWS=6 COLS=72> </TEXTAREA>  
...
```

To add a new checkbox for a new paper/reprint request, you could arbitrarily create a *Q24_Paper_Super_Discovery_Writeup* variable and add it to the form. The perl script that processes form requests for Client.com will automatically handle the new variable.

Pattern Interpretations

Another partially-in-place feature of the Client site is a series of pattern interpretations, developed from a tutorial three-ring-binder created and published by Client. Kensho adapted the paper format to allow close-ups of the images directly imported from Client's Photo-CD. The complete image library of Client's Photo-CD is stored on-site in two directories:

2.2. Site—Specific Information For www.client-domain.org 29

~/www/graphics/base-16/ (thumbnail jpegs 2–3K each)

~/www/graphics/base/ (full-resolution jpegs 30–60K each)

At present, the complete text from the Client three-ring binder is contained in the pages under <http://www.client-domain.org/education/>. The pattern images are not correctly matched with the descriptions.

Client can easily adapt this pattern-id framework to offer a series of pattern-id's as a public service, or provide password-protected analysis of client-provided slide-images using the same template.

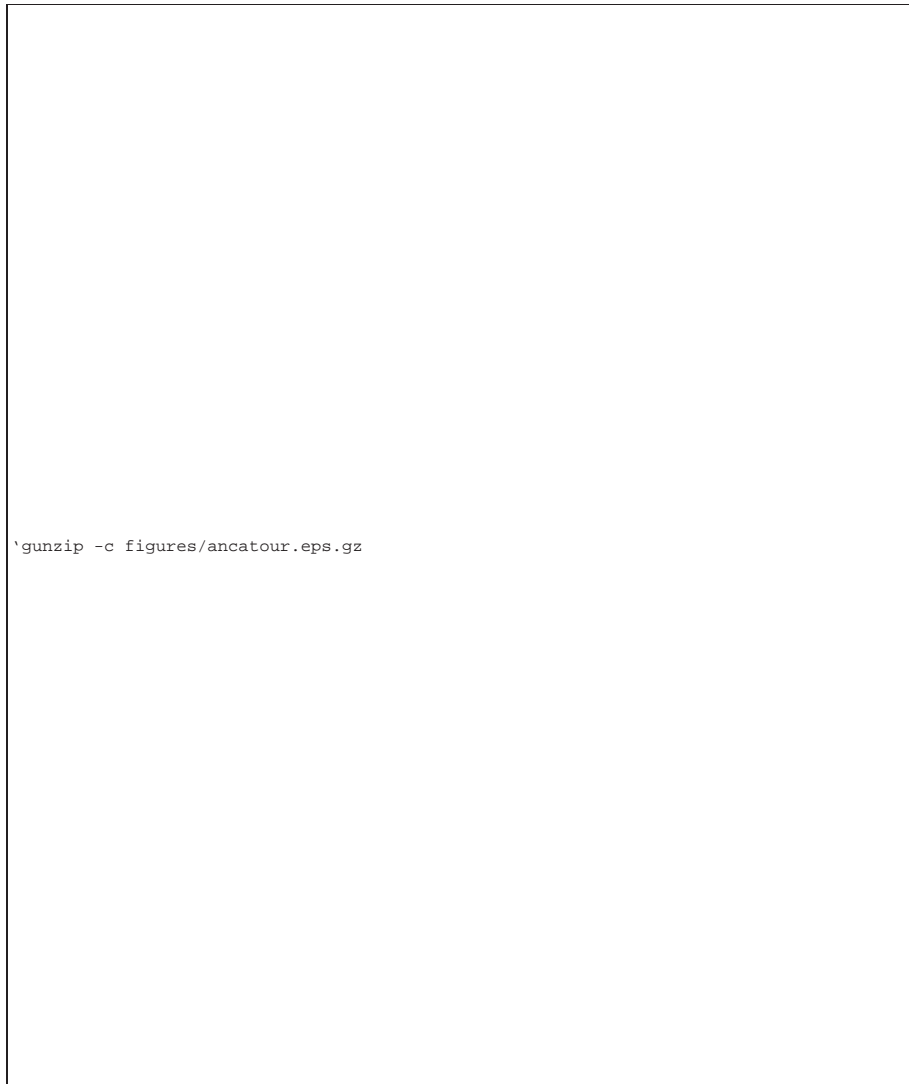


Figure 2.1: Customer Tutorial Framework

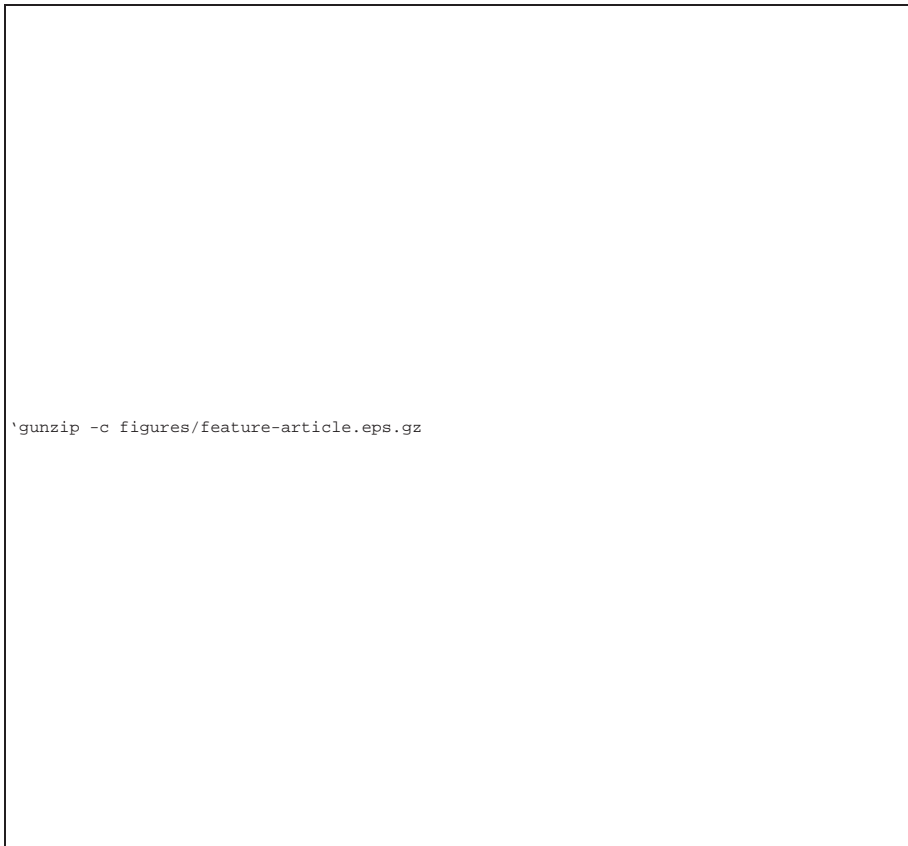


Figure 2.2: Client Home With Feature Text in Upper Right



Figure 2.3: Framework for Pattern Interpretation

Chapter 3

Tasks—How To

This guide hopes to provide a task-oriented introduction to administering the ‘virtual domain’ flavor of W^3 , mail, and other Internet information services. If you have correction, updates, or have written up a relevant task and want to see it included here, please e-mail your contribution to: admin-guide@kensho.com. ■

3.1 Connect To Your Website for Remote Administration

If you're not comfortable connecting to remote UNIX machines and manipulating files, processes, and programs, then this mini-tutorial is for you. In it, we'll go through the steps of connecting, test-driving a powerful and delicate UNIX shell environment, and then close with some sources of additional information.

3.1.1 *telnet* and *rlogin*

There are literally hundreds of ways to connect to remote machines on the Internet. For administration, we'll primarily be using *telnet* and *rlogin*, plus File Transfer Protocol (ftp) and Hypertext Transfer Protocol (http).

telnet and *rlogin* are among the oldest Internet tools, used to issue text commands and controls from your desktop to some remote machine, which may be located across the table or across the city. Both commands give you the same result, with some slight differences in speed and stability. *rlogin* is useful when that remote machine is across the table, as it uses a slightly faster and less robust protocol to communicate with the other machine. *telnet* is preferred for all of our applications, and will be used in our examples that follow.

3.1.2 Client Software

telnet is a protocol and executable program that comes in a variety of flavors. We'll describe our favorite for each platform, and give an example of connecting to another system.

UNIX

The a full-featured *telnet* command is usually built right into the operating system. By simply typing *telnet hostname* at any UNIX shell prompt, you'll arrive at your desired destination. If you're starting your site administration session from a UNIX workstation, you can probably skip the rest of this brief introduction to UNIX.



Figure 3.1: *telnet* Connection Via UNIX Shell

From a console login, or from any terminal window, issue the command

```
telnet hostname
```

where *hostname* is the fully qualified domain name of the machine you'd like to connect to. At the `login:` or `username:` prompt, provide your username on that system. This is usually no longer than eight lowercase characters. Next, provide your password at the `password:` prompt. Note that what you type will not be echoed back to you. On a traditional UNIX system, only the first eight characters of your password are checked.

Macintosh

With Macintosh machines, we prefer to use the excellent NCSA Telnet software created and maintained by the National Center for Supercomputing Applications at the University of Illinois at Urbana–Champaign. We've included details of how to obtain this utility in the software section. It comes in both PowerMac and 68xxx varieties.

MS–Windows

For the various flavors of Microsoft Windows, we're currently using a shareware package called Win/QVT. It is available for both 16 and 32-bit implementations of windows. Details for obtaining the software are included in the software section of this document.

To initiate a *telnet* connection via QVT, start the program toolbar and click on the terminal icon, which appears on the far left side of the toolbar. This will bring up the start terminal session dialog box, where you can select a hostname,

or simply type one in the textbox labeled: Host Name or IP Address. Choosing OK should initiate a *telnet* window connected to the host you selected, where you can log in and begin working.



Figure 3.2: *telnet* Connection Via QVT/Net

3.2 UNIX Housekeeping

Once connected to the computer that hosts your Internet service(s), you'll need to be aware of a few tips and tricks that will save time and more importantly, keep you out of trouble—you wouldn't want to erase your entire W^3 site by misplacing a * character on the command line, now would you?

3.2.1 *rm -rf, **, and Other Scary Stuff

Yes, you can wipe out years worth of work with a single misplaced * character. Please learn from the experience of others and don't test this theory for yourself. Unlike many end-user operating systems, unlinking (deleting) a file in most flavors of UNIX will erase it permanently. Especially on those systems set up to conform with the US government's C2 (orange-book) security standards. Our weapons against über-removal are to use the *rm* command with great care, take advantage of the enhanced C-shell's protection against *rm -rf ** commands, and use revision control as often as possible. The only hope after a mis-typed command like this is tape backup, and we have no assurances of regular backups from any of our ISP's!

3.2.2 Tildes

On a more amusing note, you may already be familiar with tildes (~). Before the advent of the world wide web, the tilde character was a well-kept secret among UNIX users. Of course, now we hear it on the telephone every day as business people, scholars, and kids alike are barking URL's into the receiver "just surf over to H-T-T-P-colon-slash-slash-netcom-dot-com-tilde-suzy-slash." Well, W^3 regulars know that the tilde refers to the web-

published homepage of a given user on a given host. Before the W^3 adopted the tilde, it meant almost the same thing in the UNIX shell. Your homepage. If your home directory is under `/export/home/u999/remote-users/`, you don't want to prefix every command with `/export/home/u999/remote-users/mylogin`. So instead, most shells support tilde-expansion, which means that `~mylogin` will be automatically expanded by the shell into `/export/home/u999/remote-users/mylogin` before the operating system deals with it.

3.2.3 Filename Completion

Oh! And for people who are as averse to typing as yours truly, filename-completion is a Good Thing™. With the advanced shell configurations we employ, you can usually just type the first few characters of a filename, directory name, or program name. Then tap the `<tab>` key on your keyboard to automatically expand the rest of the word without typing!

3.2.4 *motd*—Daily News, ISP-style

The system administrators of a host computer often put important system status information in the text file `/etc/motd`. (MOTD would be standing for 'Message of the Day' in this case). It's a good idea to review this file on a regular basis, even e-mail it to yourself if you usually hang out on a different system. To review the file while online, simply type `more /etc/motd`.

3.2.5 E-mail

While most of us at Kensho are using some sort of popmail client on our desktop workstation, the need to send and receive mail from an Internet Services

host may arise. For example, to send yourself the Message of the Day from an Internet Services host that you've logged into, type this:

```
mail yourname@yourdomain < /etc/motd
```

This will start a mail message addressed to *yourname@yourdomain*, and then redirect the standard input for the message from the file */etc/motd*.

For more elaborate server-side e-mail needs, we recommend using a sophisticated mail user agent such as *pine*. We'll provide the startup commands here, please use the *man pine* command to read the comprehensive manual pages while online.

Before using *pine* for the first time, you may want to create a *~/mail* directory and symbolically link it to *~/Mail*. This will allow you to use various e-mail software applications, some of which search for your e-mail in a directory called *mail*, and others that look in a directory called *Mail*. Here's the commands you'll need to make the symlink:

```
cd ~; mkdir mail; ln -s mail Mail
```

After this, you're free to start *pine* by simply typing: *pine*.

3.2.6 Free Space

Most of our Internet Server accounts are restricted in the amount of space they may use. To check the amount of space used by an account or directory, use the *du* command. To summarize the amount of disk space used by a single directory, issue this command:

```
du -s ~
sunclub@durak:~\$ du -s ~
7956    /home/sunclub
```

This tells us that this user's homespace is occupied by almost eight megabytes of files.

To get a summary of the size taken by a single directory full of files, try this:

```
du -s directoryname

sunclub@durak:~\$ du -s public_html/
3155    public_html
```

Here we can see that the user's world-wide-web directory contains a little over three megabytes of files.

Finally, if you need to weed out especially large files, you might try the `du` command without the `-s` parameter, like this:

```
sunclub@durak:~\$ du public_html/
15      public_html/cgi/RCS
61      public_html/cgi
15      public_html/lens/RCS
32      public_html/lens
33      public_html/src
683     public_html/graphics/photocd1/big
684     public_html/graphics/photocd1
730     public_html/graphics/photocd2/big
731     public_html/graphics/photocd2
232     public_html/graphics/tifs
2079    public_html/graphics
```

```
10      public_html/RCS
15      public_html/template
```

A quick glance at this output excerpt tells us that the `graphics` directory might be a good place to begin looking for extra-large files.

To determine how we're doing against any filesystem quotas imposed by the ISP, run the `quota` command with the `-v` parameter to print a summary:

```
inova@millennianet:~> quota -v
Disk quotas for inova (uid 1010):
Filesystem      usage  quota  limit    timeleft  files  quota  limit    timeleft
/home           16025 19456 20480                1098    0     0
inova@millennianet:~>
```

3.2.7 ISP Newsgroups

While you can usually find the most urgent status information in the `/etc/motd` file, other tips, tricks, and issues are discussed in an Internet Service Provider's local UseNet Newsgroups. This is also where we can voice our opinions on policy issues and request help with complex configuration issues. Our current derivation of the *smartlist* e-mail software was recommended via an ISP newsgroup, and participation in a different thread helped avert an ISP switching over to ftp-only site administration.

If you have a direct dialin connection to a given ISP, you can probably participate in these ISP newsgroups using an enhanced GUI newsreader such as *Free Agent* for Windows, *xrn* for UNIX, or *NewsWatcher* for the Macintosh. If not, we recommend the *tin* client for UNIXterminal newsreading access. To start *tin*, simply type `tin` on the command line. You'll probably find it pretty self explanatory. Use the arrow keys to navigate, the brief text menu at the

bottom of the screen to help you figure out subscribing, unsubscribing, and posting messages. We're not going to go into an essay on newsreading, but here's a sample *tin* screen for your familiarization:

```

Group Selection (18)                                h=help
  1      1  comp.admin.policy                        Discussions of site administration pol
  2     26  comp.infosystems                          Any discussion about information syste
  3      7  comp.infosystems.gis                      All aspects of Geographic Information
  4      7  comp.infosystems.gopher                  Discussion of the gopher information s
  5          comp.infosystems.wais                  The Z39.50-based WAIS full-text search
  6      4  comp.infosystems.www                      The World Wide Web information system.
  7      1  comp.lang.perl                            Discussion of Larry Wall's Perl system
  8     71  comp.mail.mime                            Multipurpose Internet Mail Extensions
  9    103  comp.mail.sendmail                       Configuring and using the BSD sendmail
 10     81  comp.os.linux.announce                   Announcements important to the Linux c
 11          comp.os.ms-windows.nt
 12     20  comp.protocols.kerberos                  The Kerberos authentication server.
 13    181  comp.protocols.ppp                       Discussion of the Internet Point to Po
 14    182  comp.protocols.tcp-ip                    TCP and IP network protocols.
 15      1  comp.security.announce                   Announcements from the CERT about secu
 16   577  comp.unix.solaris                          Discussions about the Solaris operatin

```

<n>=set current to n, TAB=next unread, /=search pattern, c)atchup,
 g)oto, j=line down, k=line up, h)elp, m)ove, q)uit, r=toggle all/unread,
 s)ubscribe, S)ub pattern, u)unsubscribe, U)nsu pattern, y)ank in/out

3.2.8 Shell Configuration

When you log into your account on the Internet Service host via *telnet*, you're sitting at something called a UNIX shell prompt. The *shell* is a program that interprets everything you type and passes the information along to the operating

system to act on it. Most Kensho Internet Services are configured and administered using one of two advanced shells—the enhanced C-shell (*tcsh*) and the Bourne–Again SHell (*bash*). Although you can switch to any other shell (*ash*, *cs**h*, *ksh*, *sh*, *zsh*, etc.), we strongly recommend that you leave the shell alone.

tcsh

At startup, your t-shell will follow the instructions in the `~/.cshrc` file. Kensho usually adds the following lines to this startup file, which provide some protection against accidentally erasing everything, create a recognizable user prompt, turn on filename expansion, safe file permissions, prevent creation of large core files, and prompt for your password or log you out after a reasonable amount of inactivity. Other options can be gleaned from the `man tcsh` and `man csh` manual pages.

```
## .cshrc
set promptchars = "%#" # Default and root prompt chars
set prompt = "%b%n@m%b:%~%# " # Prompt me with user@host%
set rmstar # For rm * protection
set autolist # List possible filenames w/tab
set autologout = ( 36 12 ) # Autologout, Password Check
umask 027 # Share files with group only
limit coredumpsize 0 # Maximum size of core files created
```

bash

At startup, a *bash* shell follows instructions in the `~/.profile` file. Kensho usually adds the following lines to this startup file, to create a recognizable user prompt, safe user permissions, no core dumps, and to log you out after a reasonable

amount of inactivity. More features can be easily adopted from the `man bash` and `man sh` manual pages.

```
## .profile
PS1='\u@\h:\w\$ `# Prompt me with user@host$
TMOUT=2160 # Autologout if 36 minutes idle
umask 027 # Share files with group only
ulimit -c 0 # Maximum size of core files created
```

3.2.9 More Information

This section of the manual barely touches on the UNIXism's you'll need to be aware of and comfortable with to successfully maintain and improve a client's Internet services. Further exploration of the following resources is highly encouraged.

UNIX Primer and Manpages

Garfinkel, Simson, Weise, Daniel, and Steven Strassmann, eds. *The UNIX-HATERS Handbook*. IDG Books, Programmers Press, 1994.

man and *xman*

Shell Configuration Primers

`man bash`

`man tcsh`

3.3 Change an Existing HTML Page

What procedure should you follow to make a minor change in an existing W^3 page? Once you've connected to the Internet Service host via *telnet*, follow these steps:

3.3.1 Revision Control System Checkout

Our files are stored as read-only by the gnu revision control system. Before you can save any changes to a file, you must first check it out and lock the file for editing, using the `co -l` command. In our example, we'll be changing a file called `prizes.htm`.

```
sunclub@durak:~/public_html$
sunclub@durak:~/public_html$ co -l prizes.htm
RCS/prizes.htm,v -> prizes.htm
revision 1.1 (locked)
done
sunclub@durak:~/public_html$
```

3.3.2 Edit in *joe*

Once you've checked a file out of the revision control system for editing, you can use a number of tools and utilities for editing or replacing it. In our examples, we'll be using *joe*, a freeware text editor that is a clone of the old WordStar word processor. To edit the file, simply type `joe filename`, where `filename` is the name of the file you'd like to edit.

```
sunclub@durak:~/public_html$ joe prizes.htm
```

The file will appear in the editor window. Note the Revision control information contained in an HTML comment near the top of the file.

```
IW prizes.htm Row 1 Col 1 1:41 Ctrl-K H for help
<!DOCTYPE html PUBLIC "-//W3O//DTD W3 html3.0//EN" "html.dtd">
<html><HEAD>
<TITLE>Prizes from The Sunglasses Club </TITLE>
<META NAME="KEYWORDS" CONTENT="the sunglasses club sunglass shades specs spe
<META NAME="DESCRIPTORS" CONTENT="">
<!-- Revision Control Information
-$Revision: 1.12 $
-$RCSfile: vsag.tex,v $
-$Author: sean $
-$Date: 1997/09/02 21:02:49 $
-$Id: vsag.tex,v 1.12 1997/09/02 21:02:49 sean Exp sean $
->
</HEAD>
<BODY BACKGROUND=/graphics/bg-001a.jpg BGCOLOR="#FFefd1" TEXT="#000000" LINK="#"
<BASEFONT SIZE=3>
<CENTER>
<!-- Insert Image and Repeat Title -->
<IMG SRC="/graphics/prizes.gif" align=right border=0>
<H1>

** Joe's Own Editor v2.8 ** Copyright c 1995 Joseph H. Allen **
```

Unlike the traditional vi editor in UNIX, joe is somewhat intuitive. To view onscreen help, type <ctrl>-K-H. To save the file and quit, type <ctrl>-K-X.

3.3.3 *weblint*

To assure that your revised HTML file adheres to the current HTML standard, you can check it for ‘lint’ using a *perl* utility called *weblint*. In general, just type *weblint filename* and pay attention the diagnostic output of the program. In our example site, we’ve used some of the Netscape–enhanced markup, so we’ll invoke *weblint* with a `-x netscape` parameter to accept those extensions as valid HTML.

```
sunclub@durak:~/public_html$ weblint -x netscape prizes.htm
prizes.htm(21): IMG does not have ALT text defined.
prizes.htm(38): empty container element <P>.
prizes.htm(42): tag <html> should only appear once. I saw one on line 2!
prizes.htm(0): no closing </html> seen for <html> on line 2.
prizes.htm(0): no closing </html> seen for <html> on line 42.
sunclub@durak:~/public_html$
```

In this case, *weblint* caught an important error with the inconsistent `<HTML>` tags, as well as some other details that we went back and fixed right away. After these repairs, here’s the output of *weblint* run on the same file:

```
sunclub@durak:~/public_html$ weblint -x netscape prizes.htm
sunclub@durak:~/public_html$
```

3.3.4 Test In Browser

Another important test after changing an HTML file is to review it in a *W³* browser. We’re going to assume you can figure this one out on your own J.

3.3.5 *rcs* Checkin

When the revised file meets with your approval and passes the *weblint* test, use the revision control system to check in the locked version of the file and check out a read-only one in its place. *rcs* will prompt you to enter a brief summary of the changes you've made to the file, which you can enter on one or more lines and then finish by typing `.` on a line by itself.

```
sunclub@durak:~/public_html$ ci prizes.htm; co prizes.htm
RCS/prizes.htm,v <- prizes.htm
new revision: 1.2; previous revision: 1.1
enter log message, terminated with single '.' or end of file:
>> added this week's prize
>> fixed inconsistent html tag
>> added an ALT tag for the prizes gif
>> .
done
RCS/prizes.htm,v -> prizes.htm
revision 1.2
done
sunclub@durak:~/public_html$
```

The read-only file should now be available on the world wide web, and you'll be able to revert to the previous version if needed.

3.3.6 More Information

You'll find links to software and documentation on *rcs*, text editors, HTML checkers, web browsers, and more at the end of this manual.

3.4 Transfer Files To And From Your Site

While the instructions for in-place editing offered earlier may prove useful for ‘quick and dirty’ online changes, we often need a way to package and transfer large collections of files after making major changes and improvements. If your connection to the Internet is via telephone line, you’ll appreciate the online costs that can be saved by making major revisions locally, compressing, uploading, and then de-compressing the resulting package.

3.4.1 *ftp* Software Options

This uploading and downloading, or file-transfer, has long been available as an Internet Service known as ftp: File Transfer Protocol. We’ll demonstrate here how to get and put large collections of files using various ftp clients, plus a note on how standard W^3 browsers can do part of this task. ftp software utilities vary by platform, we’ll review a few.

UNIX Command Line *ftp*

When working in the UNIX environment, Kensho recommends the enhanced ncftp client for transferring files. In the figure below, we get a client’s compressed website package using the ncftp client, then decompress the package with a combination of the gnu tar and gunzip programs.

Fetch

Fetch is a graphical file transfer tool for the Macintosh out of Dartmouth College. First time users will enjoy the animated puppy-agent who ‘fetches’ your



Figure 3.3: File Transfer with *ncftp*

files to and from cyberspace. Kensho offers a link to fetch on the training page: <http://www.kensho.com/training/>.

ws_ftp

Winsock-ftp is an excellent free utility available for several flavors of the Microsoft Windows OS and environment. Kensho offers a link to *ws_ftp* on the training page: <http://www.kensho.com/training/>



Figure 3.4: File Transfer With *ws_ftp*

W³ Browser Client (downloading only)

When you are working on W³ sites, it may be convenient to get files from a particular user account without exiting your browser. To do this, simply point the browser to the URL of the user, using ftp protocol, like this:

`ftp://username@hostname/`

In the example captured below, we're logging in to grab files from this URL:

`ftp://kathy@durak.kensho.com/`

Unfortunately, the http protocol only provides for downloading files via ftp, so you'll have to be handy with some of the other tools when you're ready



Figure 3.5: *ftp* Login Via *Netscape*

to put things back in place.

3.4.2 Compression Software Considerations

Similar to the way there are many flavors of ftp utility, there are myriad means to squish your files from large to small, many to one. Kensho prefers to work with three: gnu tar, GNU gzip, and INFO–zip. These three powerful tools are

freely available for all the major computing platforms.

tar

tar, the original UNIX tape archiving utility, has been developed by the gnu project into an excellent tool for combining many directories and files into one archive file. *tar* isn't a compression tool per-se, it's more of an advanced way to selectively & recursively archive files. *tar* can be transparently combined with GNU *gzip* to quickly create small and portable archives. To do this, use the `-z` parameter to filter the *tar* archive through GNU *gzip*. In the excerpt below, we *tar* and *gzip* the contents of the directory `www` into a single compressed archive named *inovaweb-2.0.tgz*.

```
sean@durak:~$ tar -czf inovaweb-2.0.tgz www
sean@durak:~$ ls -l inovaweb-2.0.tgz
-rw-r-r-  1 sean      support    3095122 Feb 20 14:47 inovaweb-2.0.tgz
sean@durak:~$ du -s www
9724      www
sean@durak:~$
```

The *ls* and *du* information reveal that we've succeeded in creating a three-megabyte archive from a nine-megabyte directory hierarchy.

Here's a brief summary of *tar* parameters:

```
tar: You must specify one of the r, c, t, x, or d options
Try 'tar --help' for more information.
sean@durak:~/documents/vsag$ tar --help
Usage: tar [OPTION]... [FILE]...
Mandatory or optional arguments to long options are mandatory or optional
for short options too.

Main operation mode:
  -t, --list                list the contents of an archive
  -x, --extract, --get      extract files from an archive
```

```

-c, --create          create a new archive
-d, --diff, --compare find differences between archive and file system
-r, --append         append files to the end of an archive
-u, --update         only append files newer than copy in archive
-A, --catenate       append tar files to an archive
--concatenate        same as -A
--delete             delete from the archive (not on mag tapes!)

```

Operation mode modifiers:

```

-W, --verify         attempt to verify the archive after writing it
--remove-files       remove files after adding them to the archive
-k, --keep-old-files don't overwrite existing files from archive
-S, --sparse         handle sparse files efficiently
-O, --to-stdout      extract files to standard output
-G, --incremental   handle old GNU-format incremental backup
-g, --listed-incremental handle new GNU-format incremental backup
--ignore-failed-read do not exit with non-zero on unreadable files

```

Handling of file attributes:

```

--atime-preserve     don't change access times on dumped files
-m, --modification-time don't extract file modified time
--same-owner         create extracted files with the same ownership
-p, --same-permissions extract all protection information
--preserve-permissions same as -p
-s, --same-order     sort names to extract to match archive
--preserve-order     same as -s
--preserve           same as both -p and -s

```

Device selection and switching:

```

-f, --file=[HOSTNAME:]FILE use archive file or device FILE on HOSTNAME
--force-local           archive file is local even if has a colon
--rsh-command=COMMAND use remote COMMAND instead of rsh
-[0-7][lmh]            specify drive and density
-M, --multi-volume     create/list/extract multi-volume archive
-L, --tape-length=NUM  change tape after writing NUM x 1024 bytes
-F, --info-script=FILE run script at end of each tape (implies -M)
--new-volume-script=FILE same as -F FILE

```

Device blocking:

```

-b, --block-size=BLOCKS block size of BLOCKS x 512 bytes
--block-compress       block the output of compression for tapes
-i, --ignore-zeros     ignore blocks of zeros in archive (means EOF)
-B, --read-full-blocks reblock as we read (for reading 4.2BSD pipes)

```

Archive format selection:

```

-V, --label=NAME       create archive with volume name NAME
-o, --old-archive, --portability write a V7 format archive (not ANSI)
-z, --gzip, --ungzip   filter the archive through gzip
-Z, --compress, --uncompress filter the archive through compress
--use-compress-program=PROG filter through PROG (must accept -d)

```

Local file selection:

```

-C, --directory DIR    change to directory DIR
-T, --files-from=NAME  get names to extract or create from file NAME
--null                 -T reads null-terminated names, disable -C

```

```

--exclude=FILE          exclude file FILE
-X, --exclude-from=FILE exclude files listed in FILE
-P, --absolute-paths    don't strip leading '/'s from file names
-h, --dereference        dump instead the files symlinks point to
-l, --one-file-system    stay in local file system when creating archive
-K, --starting-file=NAME begin at file NAME in the archive
-N, --newer=DATE         only store files newer than DATE
--after-date=DATE       same as -N

Informative output:
--help                  print this help, then exit
--version               print tar program version number, then exit
-v, --verbose           verbosely list files processed
--checkpoint            print directory names while reading the archive
--totals                print total bytes written while creating archive
-R, --record-number     show record number within archive with each message
-w, --interactive       ask for confirmation for every action
--confirmation         same as -w

```

On *this* particular tar, the defaults are -f - and -b 20.

For more information, type `man tar` at the shell prompt.

GNU Zip

With *gzip*, the gnu project has made big improvements over the original UNIX compress utility, eliminating copyright conflicts in the process. Remember that *gzip* is designed to compress individual files, rather than hierarchical collections of files and directories. *gzip* is best used in conjunction with tar. In the example below, we apply *gzip* compression to a tar archive we've just created, and run `ls` to view the space saved by compression.

```

sean@sophomore:~$ tar -cf mypage.tar homepage/
sean@sophomore:~$ ls -l mypage.tar
-rw-r-r-  1 sean      support      20480 Feb 20 14:50 mypage.tar
sean@sophomore:~$ gzip -9v mypage.tar
mypage.tar:          39.9% - replaced with mypage.tar.gz
sean@sophomore:~$ ls -l mypage.tar.gz
-rw-r-r-  1 sean      support      12337 Feb 20 14:50 mypage.tar.gz

```

```
sean@sophomore:~$
```

Here's a brief summary of *gzip* parameters:

```
gzip 1.2.4 (18 Aug 93)
usage: gzip [-cdfhlLnNrtvV19] [-S suffix] [file ...]
  -c -stdout          write on standard output, keep original files unchanged
  -d -decompress     decompress
  -f -force          force overwrite of output file and compress links
  -h -help           give this help
  -l -list           list compressed file contents
  -L -license        display software license
  -n -no-name        do not save or restore the original name and time stamp
  -N -name           save or restore the original name and time stamp
  -q -quiet          suppress all warnings
  -r -recursive      operate recursively on directories
  -S .suf -suffix .suf use suffix .suf on compressed files
  -t -test           test compressed file integrity
  -v -verbose        verbose mode
  -V -version        display version number
  -1 -fast           compress faster
  -9 -best           compress better
file...            files to (de)compress. If none given, use standard input.
```

For more information, type `man gzip` at the shell prompt.

INFO-zip

The *INFO-zip* project hosted by the nasa's Jet-Propulsion lab maintains a *pkzip*-compatible compression tool for many platforms, including mac, dos, unix, 16, and 32 bit windows programs. *Info-zip* combines the compression of *gzip* with the extensive file-selection options of *tar*.

Here's a brief summary of *zip* parameters:

```
Zip 2.0.1 (Sept 18th 1993). Usage:
zip [-options] [-b path] [-t mmddyy] [-n suffixes] [zipfile list] [-xi list]
The default action is to add or replace zipfile entries from list, which
can include the special name - to compress standard input.
If zipfile and list are omitted, zip compresses stdin to stdout.
-f  freshen: only changed files      -u  update: only changed or new files
-d  delete entries in zipfile        -m  move into zipfile (delete files)
-k  simulate PKZIP made zipfile      -g  allow growing existing zipfile
-r  recurse into directories         -j  junk (don't record) directory names
-0  store only                       -l  convert LF to CR LF (-ll CR LF to LF)
-1  compress faster                  -9  compress better
-q  quiet operation                  -v  verbose operation
-c  add one-line comments            -z  add zipfile comment
-b  use "path" for temp file          -t  only do files after "mmddyy"
-@  read names from stdin             -o  make zipfile as old as latest entry
-x  exclude the following names      -i  include only the following names
-F  fix zipfile (-FF try harder)     -D  do not add directory entries
-T  test zipfile integrity            -L  show software license
-y  store symbolic links as the link  instead of the referenced file
-h  show this help                    -n  don't compress these suffixes
```

For more information, type `man zip` at the shell prompt.

Freeze, StuffIt, Compact Pro, BinHex, Zoo, Arj, Lzh, Boo

A number of other compression and archive tools are popular on various computing platforms, especially those like Macintosh, where the HFS filesystem stores each file as a trio of resource, data, and finder information forks.

To maintain cross-platform portability, Kensho uses only the *tar*, *gzip*, and *zip* tools described above.

3.4.3 More Information

The *ncftp* package is available from: <ftp://ftp.cs.unl.edu/pub/ncftp/>

3.5 Change The ‘Automatic Overnight’ Stuff

Most Internet Information services do some automated housekeeping—every night files are re-indexed, archives recreated, statistics mailed. Stuff you can do by hand every day but probably don’t want to J.

UNIX provides an excellent scheduling facility, known as the cron daemon. Throughout the night (and day), cron looks to see if you have any automated jobs to do. An automated job can be anything from a single command line to an elaborate shell script. Your instructions to the cron daemon are contained in a table-file of commands for your account known as your crontab.

3.5.1 The *crontab* Command

crontab varies by system. Be very careful before invoking the *crontab* command, as certain flavors will overwrite your existing schedule by default! With that out of the way, there are two basic ways to invoke the *crontab* command.

To list (review) your scheduled table of jobs, type `crontab -l`.

```
seand@bitmed:~% crontab -l
#
0 6 * * * /export/home/website/special-bin/search-index/makeindex.sh█
#
seand@bitmed:~%
```

To edit your schedule of jobs, type `crontab -e`. This will invoke your preferred editor and open a copy of your *crontab* file for editing. When you save the file and exit the editor, the changes to your jobs will be added to the schedule. To change the editor that is used with `crontab -e`, change your `$editor` environment variable.

3.5.2 Suggestion: Call One Script

Even our favorite *W*³ administration book *Managing Internet Information Services??* lists acres of crontab additions, each of which performs some minute task of ftp site indexing, updating, what's new page generation, etc.

For ease of administration and testing, we recommend that your site's *crontab* file simply contain a call to just a few scripts, which are executed on a nightly, weekly, or monthly basis as needed.

Any changes to these scripts can be tested immediately, and you've done away with the danger whereby some flavors of the *crontab* command will overwrite your existing one by default!

The following crontab file is an example of this:

```
# $id: crontab,v 3.3 1995/03/11 01:46:31 sean Exp sean $■
12 6 * * * /home/username/scripts/nightly.sh
12 6 * 1 * /home/username/scripts/weekly.sh
12 6 1 * * /home/username/scripts/monthly.sh
```

This example runs the script *nightly.sh* every morning at 6:12 am, the script *weekly.sh* on 'day one' of each week at 6:12 am, and the script *monthly.sh* on the first day of each month at 6:12 am. 'Day One' varies by operating system, it may mean Monday to some and Sunday to others. Most versions of cron accept much more elaborate time specifications for running jobs. See the *crontab* manpage for the scoop.

3.5.3 Testing Before You Go Unattended

Before accepting even minor changes to your crontab, it is imperative that the new commands or scripts are thoroughly tested. A mistake in the crontab or

scripts called therein can inadvertently disable your information services, erase files, or run up your ISP bill with extra storage or CPU cycles.

3.5.4 More Information

cron daemons and the *crontab* format vary widely. It's best to check the manual pages on the system where your information service resides. Try typing `man crontab` and `man cron`, in that order, to learn about scheduling your jobs.

Another alternative for job scheduling is the *at* tool. *at* is designed to run a single command or script at a pre-specified time. For security reasons, some systems don't allow individual users to schedule *cron* jobs. In this case, you can call *at* in the beginning of shell scripts to schedule them to run and re-run. For details, type `man at`.

3.6 Re-Build the Searchable, Fulltext Website Index

After making significant changes to a website, such as adding or removing several pages, you might like to update the searchable fulltext index immediately, without waiting for the nightly script to run. Most of our sites employ *swish* (Simple Web Indexing for Humans) indexing, which is very efficient compared to wais and Glimpse.

3.6.1 Using *swish*

To reindex a site, you simply invoke the *swish* command with the appropriate configuration file as a parameter. Here's what it looks like, assuming you're not running verbose (which lists each and every file as it is indexed):

```
root@jmc:/home/website/cgi/muir-index# swish -c swish-interweb.conf
Removing very common words... no words removed.
Writing main index... 8861 unique words indexed.
Writing file index... 471 files indexed.
Running time: 1 minute, 46 seconds.
Indexing done!
root@jmc:/home/website/cgi/muir-index#
```

On most Kensho-administered websites, we keep the *swish* binary and configuration file in the *cgi-bin* directory, since *swish* may be used as a cgi application.

3.6.2 More Information

Swish is yet another excellent tool developed by EIT's Kevin Hughes. Invoking `swish -h` provides a fast overview:

```
sean@durak:\~\{\}\$ swish -h
usage: swish [-i dir file ... ] [-c file] [-f file] [-l] [-v (num)]
           swish -w word1 word2 ... [-f file1 file2 ...] [-m num] [-t str]
           swish -M index1 index2 ... outfile
           swish -D file
           swish -V
options: defaults are in brackets
-i : create an index from the specified files
-w : search for words "word1 word2 ..."
-t : tags to search in - specify as a string
     "HBthec" - in head, body, title, header,
     emphasized, or comments
-f : index file to create or search from [index.swish]
-c : configuration file to use for indexing
-v : verbosity level (0 to 3) [0]
-l : follow symbolic links when indexing
-m : the maximum number of results to return [40]
-M : merges index files
-D : decodes an index file
-V : prints the current version
```

The configuration file is well-documented but a little tricky with all the mandatory re-mapping of directory paths into URL's. If a site is working, we discourage messing with the config file! The official *swish* documentation is available from EIT's website at this URL: <http://www.eit.com/software/swish/>.

3.7 Create An Electronic Mail Discussion List

For small, centrally administered electronic mail discussion lists, Kensho uses the a simple *procmal* recipe to forward messages to multiple recipients. For full-scale, bulletproof lists, Kensho uses the *smartlist* package, which is an elaborate extension of *procmal*.

3.7.1 Questions to Ask Yourself

An electronic mail list may require some significant time and effort to administer, moderate, and keep active. The *smartlist* tool will take care of the boring repetitive tasks, but you may still want to think twice before establishing a list for a client.

Do We Need to Limit Access to This List?

If a list is for internal purposes only, you may choose to limit access via some built-in features of *smartlist*. Always remind yourself and our client that Internet e-mail is in no way private or secure unless messages are scrambled using some heavy duty cryptography, such as Phil Zimmerman's Pretty Good Privacy(PGP).

How Many People Will Correspond Using This List?

If you're just setting up a temporary or small-scale list, you might also find the discussion in the virtual e-mail/*procmal* section useful. That's where we give an example of a simple, centrally administered list using a *procmal* recipe.

If the answer is more than about 10 people, or if the list will include remote users for whom the list administrator can not easily provide support, the list should be set up using *smartlist*, rather than a *procmail* recipe. This will spare the list administrator many hours. The rest of this section describes how to create and configure a *smartlist* e-mail list within a Kensho-maintained virtual domain.

3.7.2 *createlist*

When you're ready to create a *smartlist* e-mail list, use the *createlist* command. The command, along with all other *smartlist* e-mail commands, should be in the \$path for the administrative user's shell account. If not, the standard location in the Kensho set up is:

```
~/smartlist/.bin/
```

In the example that follows, we create a discussion list called **info-dist** to be remotely administered by the user at e-mail address **seand@bitmed.ucsd.edu**. The syntax for the *createlist* command is:

```
createlist listname maintainer@domain
```

As you can see, we used it to start the **info-dist** list below.

```
SMARTlist@sophomore:\~{}\$ createlist info-dist seand@bitmed.ucsd.edu
Installed the following files (many hardlinked):
drwxrws--x   3 list      list          512 Mar  9 21:56 info-dist
-rw-rw-r--   2 list      list           62 Mar  9 21:56 info-dist/accept
drwxrws---   3 list      list          512 Mar  9 21:56 info-dist/archive
```

```

-rw-r--r--  4 list  list           629 Jul 26 1995 info-dist/archive.txt
drwxrws---  2 list  list           512 Mar  9 21:56 info-dist/archive/latest
-rw-rw-r--  2 list  list            62 Mar  9 21:56 info-dist/dist
-rw-r--r--  4 list  list          2189 Jul 26 1995 info-dist/help.txt
-rw-rw----  1 list  list          4077 Mar  9 21:56 info-dist/rc.custom
-rw-r-----  4 list  list         10047 Jul 26 1995 info-dist/rc.init
-rw-r--r--  4 list  list          8060 Jul 26 1995 info-dist/rc.request
-rw-r--r--  4 list  list          6351 Jul 26 1995 info-dist/rc.submit
-rw-rw----  4 list  list            0 Jul 26 1995 info-dist/reject
-rw-r--r--  4 list  list           351 Jul 26 1995 info-dist/subscribe.txt
-rw-r--r--  4 list  list           258 Jul 26 1995 info-dist/unsubscribe.txt

```

Now make the following entries in your `/usr/lib/aliases` file:

```

#####
info-dist: "|exec /export/home/list/.bin/flist info-dist"
info-dist-request: "|exec /export/home/list/.bin/flist info-dist-request"
info-dist-dist: :include:/export/home/list/info-dist/dist
#####

```

And make sure to run `newaliases` afterwards.

```
SMARTlist@sophomore:~$
```

Since we're not running as the root user on a dedicated UNIX host, we'll have to tell the *procmail* program that filters all incoming e-mail to our virtual domain about the new list, using the three lines that *smartlist* had intended for `/etc/aliases`. We usually store all the individual *smartlist* startup information in a *smartlist.rc* file named `rc.sitename.smartlist`, kept in the `~/smartlist/` directory and also symlinked to the `~/procmail/` directory.

3.7.3 *rc.sitename.smartlist*

smartlist is designed to work for root administrators of real hardware. To make the list succeed in a virtual e-mail domain, there's a little workaround you'll need to handle the incoming e-mail. Here's how we'd modify our *procmail smartlist.rc* file to handle the example list **info-dist**.

```
#####
#info-dist: "|exec /export/home/list/.bin/flist info-dist"
#info-dist-request: "|exec /export/home/list/.bin/flist info-dist-request"
#info-dist-dist: :include:/export/home/list/info-dist/dist
#####
:0:.info-dist-request.lock
  * ^TOinfo-dist-request@kensho.com
  * !^X-Diagnostic
  * !^X-Mailing-List
  * !^From.*-request
|/export/home/list/.bin/flist info-dist-request

:0:.info-dist.lock
  * ^TOinfo-dist@kensho.com
  * !^X-Diagnostic
  * !^X-Mailing-List
  * !^From.*-request
|/export/home/list/.bin/flist info-dist
```

The first 5 lines of the file are simply a commented-out version of what *smartlist* spits out for */etc/aliases* upon creation. We keep these lines on-hand as comments in case the virtual domain is ever transferred to a real domain.

There are two recipes, one to properly handle requests to the *list-request* address, and one recipe to properly handle posts to the entire list. The recipes

just do some loop checking and then pipe it into the correct *flist* command, same as would happen via *sendmail* and */etc/aliases* on conventional hardware.

3.7.4 *rc.custom*

Each *smartlist* mail list has a dedicated file named *rc.custom*, which is used to set a number of customizations for the list. In a Kensho setup, the default location of this file will be: *~/smartlist/listname/rc.custom*. The **info-dist** example lives in *~/smartlist/info-dist/rc.custom*.

The example *rc.custom* file below is a select excerpt displaying only the variables adjusted most often.

```

IW  rc.custom                               Row 1   Col 1   9:31  Ctrl-K H for help█
maintainer      =      seand@bitmed.ucsd.edu
#LOGABSTRACT=yes      # uncomment in case of emergency█
#VERBOSE=yes         # uncomment in case of real emergency█
#size_limit       =      524288             # sanity cutoff value for submissions█
#archive_hist     =      2                 # number of messages left archived█
##restrict_archive=  yes                 # uncomment this line if you█
# want to restrict archive retrievals█
# to people on the accept list█
#foreign_submit =      yes                 # uncomment this line if you█
# want to restrict submitting to█
# people on the accept list█
##restrict_archive=  yes                 # uncomment this line if you█
# want to restrict archive retrievals█
# to people on the accept list█
##moderated_flag=    yes                 # uncomment this to make the list█
# moderated (you must create a█

```

```

# 'moderators' file for this to work)
X_COMMAND_PASSWORD =      n0-sEcuriTy      # put the password for
# X-Command mails here

** Joe's Own Editor v2.8 ** Copyright c 1995 Joseph H. Allen **

```

3.7.5 *subscribe.txt*

When a person joins a *smartlist* as a subscriber, they automatically receive a welcome message. The default message is rather curt and businesslike, you may want to customize it with some specific guidelines for the list, an FAQ, or any other information—such as how to unsubscribe :-). The text file automatically sent to each new subscriber is named *subscribe.txt* and is kept in the `~/smartlist/listname/` directory. The example below comes from `~/smartlist/info-dist/subscribe.txt`.

```

IW  subscribe.txt          Row 1   Col 1   9:30  Ctrl-K H for help

This is an automated subscription mechanism.  For your verification, a
transcript of the original subscription request is included below.  If the
wrong address has been subscribed and you seem to be unable to fix it
yourself, reply to this message now (quoting it entirely (for diagnostic
purposes), and of course adding any comments you see fit).
--

```

```
** Joe's Own Editor v2.8 ** Copyright c 1995 Joseph H. Allen **
```

3.7.6 *x-command*

smartlist allows for basic remote administration to be performed through an added e-mail header named X-command. Some e-mail programs and editors will allow you to directly edit the mail headers to add the necessary information. If the list will be administered remotely from a UNIX host, you can create an *x-command* shell script to run on the administrator's remote UNIX host.

S.R. van den Berg, the *smartlist* and *procmail* author, provides a template for creating an *x-command* for your list. All you must do is edit the file, change three lines, save it to a unique name, and make sure it is executable.

```
IW xcommand-info-dist          Row 1   Col 1   9:59  Ctrl-K H for help

#! /bin/sh
: &&O='cd .' || exec /bin/sh "$0" $argv:q # we're in a csh, feed myself to sh
$O || exec /bin/sh "$0" "$@"           # we're in a buggy zsh
#####
#      doxcommand      To send of X-Command mails      #
#                                                              #
#      Created by S.R. van den Berg, The Netherlands    #
#                                                              #
#      Customise to taste.                               #
#####

#$id: xcommand-info-dist,v 1.1 1995/03/11 01:46:31 sean Exp sean $
#
```

```
# Edit the following three definitions to suit your list█  
#  
listrequest=info-dist@sophomore.kensho.com  
maintainer=seand@bitmed.ucsd.edu  
password=n0-sEcuriTy  
#  
# In a standard environment you shouldn't need to make any changes below█  
  
** Joe's Own Editor v2.2 ** Copyright c 1994 Joseph H. Allen **█
```

3.7.7 Test the List

Before inviting the masses(or the client, or the boss, etc.) to use this wonderful new list, it's wise to run through a few tests using your own e-mail address(es) to make sure things are working as they should. At a minimum, subscribe yourself, send a test message to the list, and then unsubscribe yourself.

3.7.8 Set up Hypermail Index

If desired, you can create a public W^3 archive of list messages, sorted by thread, subject, date, etc. To achieve this, we use yet another tool from Kevin Hughes of EIT: *hypermail*. *hypermail* is designed to be called from the command line or from the `.forward` file of a dedicated user account. We haven't tested it in a *procmail* recipe yet, but it should work fine if a client needs to archive their list messages publicly.

3.7.9 Register Public Lists With PAML

If the mail list you've created is intended to generate a wide audience, you can gain international attention by mentioning your list to the husband and wife team who run the list of Publicly Accessible Mailing Lists. They'll take a short description of you list, along with instructions for subscribing and unsubscribing, and post it in the UseNet, publish it on their world wide web site. The list is published once a month.

<http://bonkers.neosoft.com/paml/>

3.7.10 Removing a List

Should the need arise, you can remove a *smartlist* mailing list by simply typing `removelist listname` at the shell prompt, where `listname` is the name of your mail list. Here's an example:

```
SMARTlist@sophomore:~$ removelist info-dist
Expunging /export/home/list/info-dist, countdown initiated:█
      3
        2
          1
            zero
Don't forget to remove the corresponding entries from█
the /usr/lib/aliases file:
#####█
info-dist:
info-dist-request:
info-dist-dist:
#####█
```

```
SMARTlist@sophomore:\~{\}\$
```

remember to edit the appropriate `.procmailrc` file or `/etc/aliases` file to complete the removal.

3.7.11 More Information

smartlist developers and maintainers participate in the high-quality *smartlist* e-mail discussion list. Here are the list addresses:

Table 3.1: SmartList Discussion Information

FUNCTION	ADDRESS
Administrative requests	SmartList-request@informatik.rwth-aachen.de
Posting messages	SmartList@informatik.rwth-aachen.de

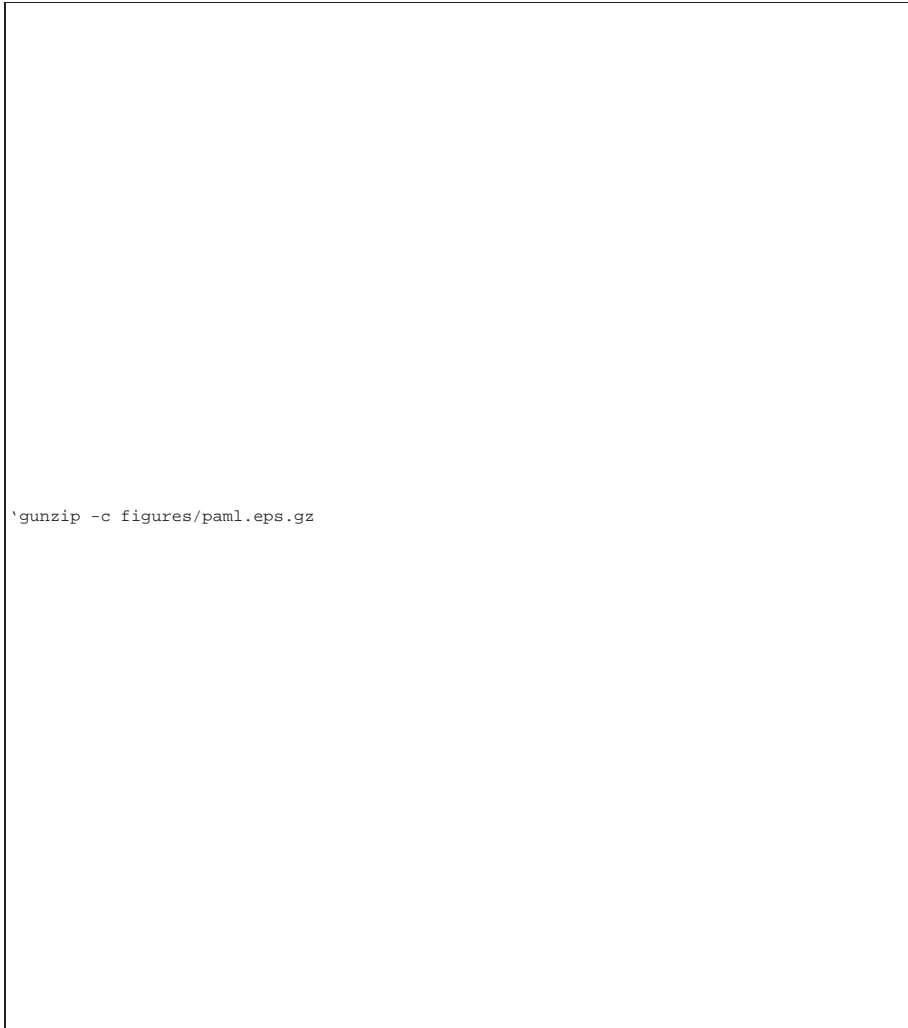


Figure 3.6: Publicly Accessible Mailing Lists Service

3.8 Working With Virtual E-mail Addresses

As an economical alternative to full-bandwidth, ‘round-the-clock’ connectivity, most Internet presence’s that Kensho sets up employ an elaborate ‘virtual’ e-mail domain.

3.8.1 What Is A Virtual E-mail Domain?

A regular e-mail address corresponds to a single user account on a host connected to the Internet. That host runs an smtp mail-transfer-agent such as sendmail to receive and deliver messages for each individual user.

When the Internet started to become super-popular beyond academia, businesses wanted the elevated status of their own Internet zone or domain, without the \$36,000 and up annual cost of maintaining a full-speed, leased-line Internet connection. As a reasonable compromise, Internet Service Providers began to offer what has come to be known as a ‘virtual e-mail domain.’

A virtual e-mail domain also corresponds to a single user account on a host connected to the Internet. That host runs an smtp mail-transfer-agent such as sendmail that is specially configured to accept mail for any user in the domain and dump it into the same single user account. Messages for dozens or hundreds of users can be dumped into this single account.

For example, our fictional client CBS might have established a virtual e-mail domain for **cbs.com** via a single user account with MillenninaNet, a local ISP. Any time anyone on the Internet addresses a piece of mail ending in **@cbs.com**, the message makes its way to the Internet host **millennianet.com**, which then dumps it into the single-user mailbox **cbs@millennianet.com**. Doesn’t matter if the message was addressed to **all-you-guys@cbs.com**, **presi-**

dent@cbs.com, cyber-monsters@cbs.com, sales@cbs.com, fred@cbs.com, and so on.

Sorting, saving, downloading, redirecting, responding, forwarding, and deleting of the mail for the entire domain is now in our hands, the ISP's responsibility has ended. Kensho relies on an excellent public-domain package called *procmail* to evaluate and process each e-mail message as it arrives. *procmail* is something akin to a cookbook that uses e-mail messages as the ingredients: it follows recipes.

3.8.2 Limitations Of Virtual E-mail Domains

Virtual e-mail domains have some immediate and obvious drawbacks:

- *procmail* can't handle messages Cc:'d within the same virtual domain. The message will reach the first user matched in the *.procmailrc* recipes, and no one else. This is why a responsible and information-literate person should appear as the first redirect-recipe—so they can recognize and reroute messages as needed.
- To achieve the illusion of a fully functional Internet domain (at least in the e-mail realm), end-users must properly configure the Reply-to: feature of their mail-user-agent software. This is an extra step that can be difficult to coordinate among remote users.
- Administration of the e-mail can become quite a daunting task, especially if the organization is large and employee turnover high.
- Running the personal messages of more than one user through a single centrally-controlled account raises some issues of privacy. The elabo-

rate and delicate *procmail* processing can occasionally misdirect messages without the slightest concern for sensitive content.

3.8.3 Configuring User Mail Agents

To solve the limitation of end-user `Reply-to:` addressing, you can help the users in your domain configure the `Reply-to:` field in their end-user mail program. To clarify our example of CBS, let's look at an imaginary user named John Muir. When *procmail* sees any message addressed to **muir@cbs.com**, it redirects the message to **cbs-02@millennianet.com**, the actual account John dials into and downloads his e-mail.

Without some minor tweaking of John's mail user agent (the program he uses to read and send e-mail), his e-mail will appear to come from **cbs-02@millennianet.com**, rather than the more prestigious **muir@cbs.com**. Replies, by default, will also be addressed to **cbs-02@millennianet.com**. Regardless of the address John keeps on his business card, folks are apt to bookmark and use whatever is automatically saved. To keep the **cbs.com** domain in circulation, Figure 3.8.3 shows how John would alter his *Eudora* (a POP mail application) configuration to use the preferred return-address.

If John was reading his e-mail with Netscape's software, he would pull down the Options menu and select the Mail and News Preferences... dialog box. In the `Reply-to Address` textbox, he'd enter the preferred **muir@cbs.com**, as shown in Figure 3.8.3.

3.8.4 Cooking With *procmail*: Recipes

As every e-mail message arrives in the virtual e-mail domain, Kensho has configured the *procmail* program to evaluate the message headers against a list of regular expressions, and act accordingly to save, forward, delete, or reply to the message. The instruction to filter every message through *procmail* is usually kept in the `~/.forward` file on an ISP machine. The line in the `~/.forward` file to invoke *procmail* varies with the location and version of *procmail*. In most cases, we've configured *procmail* 3.1x like this:

```
"|IFS=' '&&exec /homedir/bin/procmail -f-||exit 75 #username"█
```

On Kensho's hardware, we've made *procmail* the default local mail delivery agent, so all you need is a *.procmailrc* file on our system.

Recipes

procmail compares the headers and/or text of each incoming mail message against a set of processing recipes that we establish. In virtual mail domains, it is common to create a recipe for each user in the organization. When a piece of mail arrives for a particular user, *procmail* determines to which real, final destination user account the message should be forwarded.

More advanced applications of *procmail* in virtual mail domains include the imitation of mail lists and *autoresponders* (like the old BSD UNIX *vacation* program), plus departmental aliases such as **sales@mydomain**, **help@mydomain**, **marketing@mydomain**, etc.

Forward to an Individual User This is the type of *procmail* recipe you'll probably use the most often in a virtual e-mail domain. Employees and affili-

ates who qualify for a **myname@virtualdomain.org** address will need a *procmail* alias set up. Depending on the e-mail addressing scheme of the organization, an alias may look for first name, last name, initials, or some combination thereof. The example below accepts many potential guesses and misspellings of the name **sean**. Keep in mind that the regular expressions used by *procmail* are case-insensitive by default.

```
:0
* ^TO(sean|seand|sed|shawn|shaun|sean\.dreilinger)@kensho.com
!sed@netcom.com
```

In a larger virtual domain where e-mail addressing/naming schemes require stricter adherence, Table 3.8.4 suggests some potential naming policies.

Table 3.2: E-mail name/address conventions

EXAMPLE	SCHEME	NOTES
washington	last name	Needs plan for multiple employees in the same family.
gw	initials	Needs plan for employees with matching initials. gw
gwashing	8 chars	Useful in organizations where mailnames correspond to phone numbers.
gwashington	first initial last name	Unique, getting longer to type-in.
george_washington	full name with underscore	Descriptive but tedious to type. Good luck relaying.
George.Washington	full name with period	Easy to read, hard to type. Can confuse <i>procmail</i> rules to escape the periods.

Forward To Several Users (A Manual Mail List) Where simple or temporary e-mail list functions are desired, *procmail* allows convenient cre-

ation of on-the-spot mail lists and group aliases. The example below shows a list recipe we quickly set up to establish a dedicated communication channel with an important client.

```
:0
* ^TO(cerfmktg|cerfmktng|cerfplan|cerfideas|cerf|cerfnet)@kensho.com
* !^FROM_DAEMON
* !^X-Loop: kensho@best.com
!nevilleb@kensho.com, sed@netcom.com, fparker@cerf.net
```

Such ad-hoc lists are equally effective for departmental aliases, where the actual behind-the-scenes users may change frequently. If you require a more full-strength or automated mail discussion list facility, consult the *smartlist* section of this guide.

Autoresponder A final application of *procmail* that is super-useful in virtual mail domains is the ability to set up automatic responses. You may be familiar with a similar feature on regular UNIX accounts, where a person going away for vacation could run the *vacation* program. *vacation* automatically responds to incoming e-mail with a text message of your choice, usually saying something like “I’m away from my mail until June 16th, if this is urgent please contact my colleague Fred at...”

The most frequent application of autoresponders or *mailbots* as they’re occasionally called, is for the standard **info** address for a given virtual domain. The recipe below shows how anyone sending a message to **info@kensho.com** will automatically receive a text message in return. The auto-reply message contains the full text of a file named *auto.kensho.com*.

```
# Autoresponder for messages arriving to the INFO@ address
```

```

# 1/2 Forward a copy of the message to an employee's mailbox
:0 ci
  * ^TO(aoinagi-info|info|information)@kensho.com
!sed@netcom.com

# 2/2 Mail back a text file to the enquiring address.
#   The text file here is \~/procmail/auto.kensho.info
#   NOTE: procmail doesn't do \~/ tilde expansion, complete paths required
:0 A
  *!^From +kensho
  *!^Subject:.*Re:
  *!^FROM_DAEMON
  * ^TO(info|information)@kensho.com
# formail notes
#   -r = generate autoreply
#   -k = keep message body
| (formail -r -k ; \
cat /home/kensho/procmail/auto.kensho.info) \
| \${SENDMAIL -oi -t

```

There are many creative applications for *mailbots*. You can use them to distribute survey questionnaires, price sheets, electronic coupons, and other forms of information-on-demand.

3.8.5 Test the Address(es)

Before a user or client announces their new e-mail address, please test it! At a minimum, send e-mail to the virtual address and then finger the final destination address to confirm that the mailbox has just received your message. Much better to do a full-on test before announcing the address, or else the postmaster person for the virtual domain may be doing quite a bit of manual re-mailing.

3.8.6 Revision Control

Mentioned briefly as a tool for tracking modifications to HTML files, please use revision control for all changes to system configuration files such as the *procmail* rc files. It's often the fastest and most painless way to recover from mistakes in critical system configuration tasks, especially when problems arise a few days or even a few weeks after changes were made. For a step-by-step review of revision control, please re-read the section explaining how to change an existing HTML page.

3.8.7 More Information

procmail developers and users participate in the *procmail* e-mail discussion list. Table 3.3 shows the list addresses.

Table 3.3: *procmail* Discussion Information

FUNCTION	ADDRESS
Administrative requests	procmail-request@informatik.rwth-aachen.de
Posting messages	procmail@informatik.rwth-aachen.de

There are four primary manpages for *procmail*, Table 3.8.7 contains an overview of where to find them and what they cover.

Alan Stebens maintains an excellent library of *procmail* recipes, add-ins, plus two larger tools—*ackmail*, an autoresponder for individual users, and *proclint*, a *perl* tool for checking the syntax of *procmail* recipes. Stebens offers these tools to the public domain at: <http://eci2.ucsb.edu/~aks/mail/procmail-lib.html>

Table 3.4: *procmail* Documentation Summary

MANPAGE	SECTION	CONTENTS
<i>procmail</i>	(1)	<i>procmail</i> overview.
<i>procmailrc</i>	(5)	<i>procmail</i> rc file. In-depth explanation of recipes, <i>procmail</i> environment vari
<i>procmails</i>	(5)	<i>procmail</i> weighted scoring technique beyond Boolean.
<i>procmalex</i>	(5)	<i>procmail</i> rcfile examples. Excellent comments and annotations throughout.



```
\gunzip -c figures/mail-ua-eudora.eps.gz
```

Figure 3.7: Configuring *Eudora* E-mail

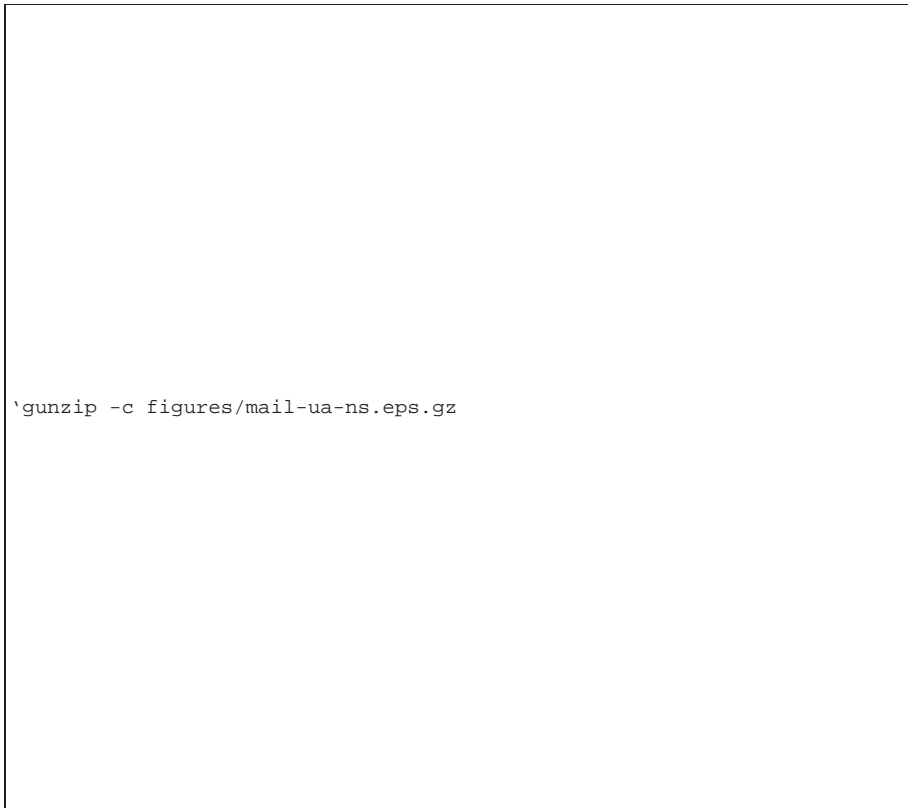


Figure 3.8: Configuring *Netscape* E-mail

3.9 Create A Password-Protected Area

Many of our W^3 netsites have a password-protected, pseudo-intranet area located in the directory *internal/* in the root of the site. This area is configured with password protection during creation of the site and assures that future administrators have a working example of a password-protected area for that account.

3.9.1 Permissions Caution!

Any time you wish to control access to your files on a shared UNIX server, *permissions* come into play. Although successful application of the techniques in this section will protect your files from unauthorized access via http, you still need to be aware of the access you are granting or denying to members of your group and other users on the UNIX host itself.

3.9.2 Procedural Overview

- Create/edit access file
- Create users and passwords
- Test the restrictions

3.9.3 Create/Edit The Access File

For smaller sites, we usually configure a top-level access and password file in the website root directory. In a virtual domain account, this is often something

like `~/public_html/` or `~/www/`. The access file should already exist unless you want to set things up outside the root of the web document tree. These are just plain old text files, joe `.htaccess` on the command line will start editing an NCSA access file in the current directory.

NCSA Web Servers: *.htaccess*

The NCSA *http* server, by default, will check each directory for a *.htaccess* file before returning information to a web browser. The *.htaccess* file may list restrictions for the current directory and any subdirectories.

Here's an NCSA *.htaccess* file configured to enforce user and group-level access starting in the current directory:

```
AuthUserFile /home/support/sean/homepage/private/fest/htdocs/.htpasswd
AuthGroupFile /home/support/sean/homepage/private/fest/htdocs/.htgroup
AuthName Fest Demo Visitor
AuthType Basic
<Limit GET>
require user sean daniel
require group Fest
</Limit>
```

A full explanation of NCSA-style Access Control is available online from the NCSA at:

<http://hoo.hoo.ncsa.uiuc.edu/docs/setup/access/Overview.html>

NOTE: Our accounts on BEST.com *W*³ servers employ a hacked *httpd* based on NCSA v1.3. This custom server looks for a *.bhtaccess* in each directory. On most BEST accounts, we've created a symlink from the *.htaccess*

file to the name `.bhtaccess`. The added `srm.conf` (resource configuration) type features available in the BEST `.bhtaccess` file are documented at this URL:

<http://www.best.com:80/FAQ/bestwwwd.html>

Netscape Web Servers: `.nsconfig`

The Netscape server, by default, will check the root web directory for a `.nsconfig` file before returning information to a web browser. The `.nsconfig` file centrally lists any access restrictions for the entire website.

Here's a central Netscape `.nsconfig` file configured to enforce user-level access control on two subdirectories. Note that the Netscape documentation requires the each directive to appear on one long line, we've formatted the example text to fit the page:

```
#####
# Netscape Server Access Config File for \CLIENT
#####
# Restrictions for the cme/ subdirectory
<Files cme/*>
RequireAuth userfile=/home/www/www.inovadx.com/.htpasswd
             realm="CLIENT CME Area Still Pending Legal Clearance"
             userlist=kensho,inova,sean,dave,kpobrien,mgarcia,neville,dave
</Files>
#####
# Restrictions for the internal/ subdirectory
<Files internal/*>
RequireAuth userfile=/home/www/www.inovadx.com/.htpasswd
             realm="CLIENT Internal"
             userlist=kensho,inova,sean,dave,kpobrien,mgarcia,neville,dave
</Files>
```


#####

The following Netscape Server documentation and tech-sheets may be helpful if you need to adjust the access control for a Netscape W^3 site:

User databases: http://home.netscape.com/comprod/server_central/config/configure.html#dbase

Access control: <http://home.netscape.com/assist/support/server/tn/cross-platform/20011.html>

NCSA/Netscape upgrade/comparison issues: http://home.netscape.com/eng/server/1.1/faq_servnca.html

A Word On (*g—n—l*)*dbm* User Lists

The current NCSA and Netscape W^3 servers can store user access lists in *dbm* database format. User lookups in *dbm*-based user files will be faster, but are slightly more difficult to maintain. We're presently using only the text-based option for user lists.

3.9.4 Create Users and Password(s)

Use the *htpasswd* utility to add or update W^3 user access information. Ken-sho has compiled the NCSA *htpasswd* utility for each virtual domain account to maintain encrypted passwords for W^3 visitors. Unless you create a cgi interface to change user passwords, these are all assigned and administered by the person maintaining the site. Here's the syntax for using the *htpasswd* command:

```
htpasswd [-c] passwordfile username
```

The `-c` flag will create a new password file (or overwrite the old one—be careful!). Below is an example in which we create a new password file and add a user named **muir**.

```
seand@interwin:~/public_html% htpasswd -c .htpasswd muir
Adding password for muir.
New password:
Re-type new password:

seand@interwin:~/public_html% more .htpasswd
muir:lRdFoAIpuPkhs

seand@interwin:~/public_html%
```

NCSA offers information for adding and managing W^3 users with the *htpasswd* command at this URL:

<http://hoofoo.ncsa.uiuc.edu/docs/setup/admin/UserManagement.html>

3.9.5 Test the Restrictions

Creating restricted user areas may disturb delicate permissions and restrictions of the existing document trees. Liberal use of revision control is strongly recommended during changes to site access control. When you think things are configured correctly, try testing both protected and unprotected areas from multiple W^3 browsers. When access problems arise, check the pathnames defined in the access configuration file first. If those are correct, make sure that the UNIX permissions on the access configuration files and password files are world-readable (`chmod 444`).

3.10 Getting Attention

Here's a question at which our clients always arrive before, during, or after the creation of an Internet information service in their name: "Once I'm out there. How will anyone ever find me?" This section of our guide will discuss a few answers to that inevitable query.

3.10.1 Coming Up First In Full-text Indices

The top ten or so Internet Indices perform full-text indexing of a given URL, W^3 site, or ftp site. In essence, after indexing by one or more of the big Internet IR engines, any words that appear on (or within) an HTML page can be retrieved by a computer (and more importantly—potential customer—that is seeking or browsing information on the Internet) searching the index. By design, any Kensho W^3 site should be easily found by those seeking the client or client site.

Browsers (people who are browsing an information subject, not to be confused with W^3 'browser' software tools) may need some artificially-assisted serendipity in order to find our client site when they express a vague interest in a relevant subject or competitor. Any time a browsing person is queries a search engine or Internet directory for a relevant subject or competitor name, electronic encouragement can be made available so that our client site will appear in many more of their search results, often near the top of any listing that might appear.

We presently improve our chances of being noticed by embedding relevant topics, competitor names, and other keywords within the actual text of the pages, within HTML comments hidden in the page, and with HTML `¡META¿` tags, each of which are read and indexed by some or all of today's web-

indexing robots.

Following are two simple examples of embedded key words. Each uses a simple `<META>` tag to suggest some alternative keywords that people might use when searching for this topic or organization. Remember that `<META>` tags must appear within the `<HEAD>` element of an HTML document.

```
<HEAD>
...
<META NAME="KEYWORDS" CONTENT="Aoinagi Karate Karate-Do Butokukai Shito-Ryu
                                Okinawa Castilonia Billimoria Richard Kim
                                UCSD Redlands">
...
</HEAD>

<HEAD>
...
<META NAME="KEYWORDS" CONTENT="BITMed bitmed biological medicine
                                medical informatics">
...
</HEAD>
```

Next is a more elaborate example, employing both HTML comments and `<META>` tags. Content appearing on viewable page is also strategically selected to produce high-recall after search-engine indexing, it uses words that will not be eliminated as ‘stop’ words during most indexing and stemming. The `<META>` tags appear at the top of the file, the HTML comments appear at the very bottom. Note we only included the long list of competitor names at the very bottom. The `KEYWORDS` tags in the HTML comments are not necessary, just thrown in to help make sense for curious people doing View-Source in their web browsers.

```
<HEAD>
...
<META NAME="KEYWORDS" CONTENT="inova autoimmune diagnostics ELISA enzyme
    linked immuno sorbent assay immuno
    flourescent ouchterlony anca">
<META NAME="DESCRIPTORS" CONTENT="inova diagnostics Inova Diagnostics
    INOVA DIAGNOSTICS">
...
</HEAD>
...
<!-- KEYWORDS = "inova diagnostics internet home page Inova Diagnostics
    Internet Home Page INOVA DIAGNOSTICS INTERNET HOME PAGE" -->

<!-- KEYWORDS = "inova autoimmune diagnostics ELISA enzyme linked immuno
    sorbent assay immuno flourescent ouchterlony anca" -->

<!-- KEYWORDS = "American Bioproducts, ABP, ALPCO, Antibodies Inc., Apotex,
    APL, ALpha Antigens, ATOM, Binding Site, Bio WHittaker,
    BMD, Bio Line, Biomedical Products, BAG, Bio Meriux, Bio
    Carb, Bio Hytech, BIOS, Bio Systems, Bio Lab, Bio Rad, Bio
    Diagnostic Ltd., Bio Allergy, Buhlmann, Cogent, Clark,
    Cheshire, Chemicon, Cormedica, Cambridge Life Sci,
    Diamedix, Digast, Diagnolab, DELTA, DPC, Euroimmune, ELIAS,
    Euro Diagnostics, Eurospital, ENI/Schiappareli, Freka
    Flour, Fresenius, General Biometrics, Gull, Genesis, Helix,
    Hemagen, HB, Hoechst Japan, Hycor, ImmunoConcepts, Immuno
    Diagnostic Products, Immunovision, INC Star, Intl. Immuno
    Diagnostics, Intl. Immunoassay, Ismunit, Immco, IBL,
    Imtec, Inova, Ken Star, Kronus, Kainos, Labor Diagnostica,
    LaboServ, Leti CBF, La Jolla Pharm, Mardx, MeDiCa, Murex,
    Miles/Seratek, MAST, MBL, Medic, Medical Innovations,
    Nichols, Orgasan, Protrack, Paeseli Lorei, Progen, Quidel,"
```

```
Reaads, Sanofi/Kallestad, Scimedix, Shield, Sigma, SDI, █  
SLR, Thera Test, Tri Delta, ViroImmune, Vira Med, Yamasa, █  
Zeus" -->  
  
</BODY>  
</HTML>
```

3.10.2 Registration and Listing

Kensho has a simple three step strategy for publicizing Internet Sites. Depending on the level of support we're giving a client, you may perform one or more of these tasks on a regular basis.

Top Ten

Each and every site we create is hand-registered with the top-ten Internet Search Indices and Directories that provide free indexing/listing. If you're performing maintenance on a site, it's your responsibility to update and test these listings on a monthly basis. Reliable means for measuring 'Internet Presence' are scarce to none, and clients (and their clients) like to play search-index roulette and see if their Internet site magically appears with their choice of search terms. All we can do is register the sites and hope they have a clear understanding of the joys and perils of Boolean logic if and when their site fails to appear J. Here's a quick rundown of the current sites we're using as our 'top ten:'

Alta Vista <http://www.altavista.digital.com/>

Comprehensive List of Sites

Excite <http://www.excite.com/>

InfoSeek (Directory and Guide) <http://www.infoseek.com/>

InkTomi <http://www.inktomi.com/>

Lycos <http://www.lycos.com/>

Open Text <http://www.opentext.com/>

WebCrawler <http://www.webcrawler.com/>

WWW Worm <http://www.wormy.com/>

Yahoo <http://www.yahoo.com/>

Listing Service

Sites who have a contract that emphasizes presence development will also be submitted to a for-hire listing service. We're presently using A1, who will place our client in the top 50 or top 100 of 600+ Internet directories, for a small gain in presence above and beyond being listed in the top ten. Please let us know if you feel its time to re-list a site that you administer.

Selected Appeals

A final, high-end service we perform is entering different information communities and making our client's site known via strategic signatures, polite plugging, and correspondence with identified information gatekeepers, often to request reciprocal links.

Strategic Signatures With strategic signatures, we participate in a UseNet discussion group or mailing list whose topic is a relevant forum to advertise our clients Internet services. One or more representatives who are knowledgeable in the subject area participate enthusiastically in the list/group traffic. At the bottom of every message, the 1–4 line signature mentions our client, their byline, and their preferred Internet address.

Polite Plugging Polite plugging is used less often, and with extreme caution to respect the Netiquette expected on the receiving end. We respond to public requests for vendors, price lists, product information with a public, promotional response which suggests “our client XYZ Inc. has some solutions you may be interested in, you can obtain further information from their world wide web site, or send a blank message to **info@clientname.org**.”

Information Gatekeepers Finally, we directly seek out and contact identified information gatekeepers for particular industries and relevant Internet sites. If the gatekeeper participates in the Internet, we’ll contact them with a polite letter of introduction (usually e–mail) and invite them to visit the site. If the gatekeeper happens to run an Internet site, we may offer a reciprocal link to our client’s page, if the client has an area for such links.

3.10.3 Notifying Existing Clients

Salespeople may enjoy the bona–fide opportunity to call on established clients and let them know about the new Internet resource(s) now available to them. An organization launching new Internet services may wish to prepare instructional materials to help clients (and employees) take advantage of the new resources.

3.10.4 Advertising

With the assistance ARK Enterprises (Kensho's print-based advertising, market-research, and consulting resource), We encourage our clients to promote their Internet resources actively in all other marketing media—radio spots, print ads, business cards, T-shirts, mugs, whitepapers, promotional literature, letterhead, product packaging, etc. This cross-media promotion can be the key to launching a successful and lasting Internet campaign. ARK Enterprises can currently be contacted as follows:

ARK Enterprises
all-ark@kensho.com
9444 Balboa Avenue Suite 235
San Diego, CA 92123
voice: +1 (619) 514-3939
fax: +1 (619) 514-3938

3.11 Adding or Changing Graphics

The advent of the W^3 has resulted in the revitalization of a few graphic file formats and features once thought to be obsolete. These features are image transparency and interlacing in the Compuserve GIF format.

3.11.1 Software

Kensho currently licenses the Windows 95 version of Micrografx Picture Publisher 6.0 for much of its W^3 -related graphics work. This is because the native Picture Publisher file format preserves textual information in an editable layer above the bitmapped image layer. Other high-end graphics packages such as Adobe Photoshop and the recently discontinued Aldus Photostyler can also support editable text layers on raster files. Being able to return to a file and edit the text saves lots of time when a client requests a change to a graphical menu or imagemap.

Where screen captures, batch image file conversion, or png support are required, we occasionally turn to the JASC Paint Shop Pro shareware package.

Finally, special public domain tools exist for creating imagemaps, and we find them most helpful for quick generation of old-fashioned .map files for server-side imagemaps. The coordinates generated can then be modified and plugged into client-side imagemaps. Kensho currently uses Todd C. Wilson's MapTHIS! for Windows NT as a freeware map editor.

3.11.2 Text

For custom–designed buttons or imagemaps, we’ll end up using lots of text within bitmapped images. Using one of the graphics packages named earlier can help make modifications to graphic–based text less tedious.

```
`gunzip -c figures/hi-national-footbar.eps.gz
```

Figure 3.9: Text–laden Navigation Bar Graphic

Font Selection

Our goal with text in bitmapped graphics is readability. Usually we use sans–serif styles such as Helvetica or Arial when working below 10 points. A life–size comparison of fonts in several *W*³ browsers is recommended before committing to any site–wide font choices.

Anti–Aliasing

Some graphics packages can anti–alias text before burning them onto the bitmap, this improves readability across many monitors and resolutions, but displays poorly on monitors that can’t produce 256 colors or shades of gray. If you know a sites clients have reasonably modern hardware, anti–aliasing can add some aesthetic appeal.

3.11.3 File Formats

There are just enough decisions to make when selecting image formats for W^3 sites to be daunting. There are two intelligent file format choices, with a third format expected to be dominant within two years.

GIF

CompuServe's Graphic Interchange format(GIF) was the first image format to be supported by W^3 browsers as an inline graphic. In early 1994, Netscape communications added the ability to process GIF 89 format interlaced and transparent images in their beta W^3 browser. All GIF images are limited to a maximum of 256 colors. The resulting file is stored in a compressed format, and there are currently copyright lawsuits and claims circulating about the compression algorithm.

Interlaced images display to the end-user as a series of partially complete horizontal lines. You see an interlaced image unfold as though it were drawn on venetian blinds that are being turned towards the monitor. The effect is most pronounced for users who access the Internet across slow modem links.

Transparent GIF 89a images specify that one of their 256 colors will be read as 'transparent' by the image viewing software, so that whatever is in the background of the computer display window will 'show through.'

JPEG

The Joint Photographic Experts Group(JPEG) defined the lossy jpeg image compression scheme as a means to store extremely large photographic image files in limited space. The jpeg format supports infinite colors and can be com-

pressed to a fraction of its source size, at the expense of image quality. jpeg depends on the receiving computer to do some hefty decompression, so we recommend its use only after a gif file has been ruled out due to excessive file size or limited color depth.

PNG

The Portable Network Graphic (PNG) file format was developed in response to the technical and legal limitations of gif and jpeg image formats. Although not supported by present-day W^3 browsers, we expect png to play a significant role in the near future.

3.11.4 More Information

Discussion of client and server-side image mapping is beyond the scope of this manual, two helpful resources are available.

Server-side Imagemaps

<http://hoohoo.ncsa.uiuc.edu/imapemap-tutorial>

Client-side Imagemaps

<http://www.netscape.com/server/client-side-maps.html>

Chapter 4

Software

This section describes software we use to construct Internet Services. Hyper-text links to obtain additional documentation, user support, and the software itself are included wherever possible.

4.1 W^3 System Administration

4.1.1 *cyclelog*

cyclelog.pl is a perl script we've adopted to rotate W^3 *access_log* files on a nightly basis. The original script was written by Martien Verbruggen, tgtdmv@chem.tue.nl. ■

Customizations

On sites where we do not own the *httpd* process, we comment out the line to `kill -1` the *httpd* *pid*:

```
die "Temporary log:$ACCESS_LOG not accessible\n" unless (-f $ACCESS_LOG); ■  
'kill -1 \$$PID';
```

Some local customization for file/binary names and locations is also required.

Local Sources

`~/bin/cyclelog.pl`

Official Distribution

<http://www.tcp.chem.tue.nl/stats/script/cyclelog.pl>

Official Documentation

<http://www.tcp.chem.tue.nl/stats/script/>

4.1.2 *getgraph*

getgraph.pl is a perl script we've adapted to generate graphical W^3 *access_log* statistics on a nightly basis. The original script was written by Martien Verbruggen, tgcmv@chem.tue.nl. *getgraph.pl* is a delicate orchestration that unites and applies a number of programs, each of which you may need to obtain and compile from scratch.

Customizations

In addition to the usual specific file/binary locations, we've added some aesthetic customizations to incorporate site-wide headers, footers, and backgrounds on all of the statistics pages.

```
# HTMLish variables added for customization
$body='<BODY BGCOLOR=#ffffff><CENTER><TABLE><TR><TD><BASEFONT SIZE=2>' ;
$footer='<BR>FOOTER GOES HERE</BASEFONT></TABLE></CENTER>' ;
```

The site-wide navigation bars are added by modifying and/or creating the following four *perl* subroutines:

```
sub print_header {...}
sub print_footer {...}
sub add_header {...}
sub add_footer{...}
```

Local Sources

~/bin/getgraph.pl

Official Distribution

<ftp://ftp.chem.tue.nl/pub/sysutils/getgraph.tar.gz>

Official Documentation

<http://www.tcp.chem.tue.nl/stats/script/>

4.1.3 *getstats*

Another Kevin HughesT creation! This command-line tool analyzes the contents of a site's *access_log* and outputs statistics in ascii text or HTML format. *getstats* is used by the *getgraph.pl* script.

Customizations

getstats can choke on standard *access_log* files unless it knows they are in common log file format. For this reason, we usually compile the program to use a common logfile format by default.

```
#ifndef COMMON
#define COMMON 1
#endif
/* Overridden by -M option.
** If your log file is in the ``common'' format, define COMMON as
** 1, else as 0.
*/
```

Local Sources

~/src/getstats.12.c

Official Distribution

<ftp://ftp.eit.com/pub/web.software/getstats/>

<http://www.eit.com/software/getstats/>

Official Documentation

<http://www.eit.com/software/getstats/getstats.html>

4.1.4 NCSA *httpd*

The leading public domain UNIX W^3 server.

Customizations

If dedicated to a specific client, we'll hard-code a number of Directive and variable values to speed things up a bit and shorten the length of the configuration files. Changes can be made to taste in the *src/* subdirectory of the distribution in these two files: *config.h* and *constants.h*.

Local Sources

We don't presently store the source for NCSA *httpd*.

Official Distribution

<http://hoohoo.ncsa.uiuc.edu/>

ftp://ftp.ncsa.uiuc.edu/Web/httpd/Unix/ncsa_httpd/

Official Documentation

<http://hoohoo.ncsa.uiuc.edu/docs/Overview.html>

4.1.5 Netscape httpd

The leading commercial W^3 server.

Official Documentation

<http://home.netscape.com/assist/support/server/index.html>

4.1.6 *swish*

The Simple Web Indexing System for Humans (*swish*) is a command-line utility to create full-text indexes of W^3 document trees. *swish* understands HTML and has a very efficient index format. Yet another indispensable resource from EIT's Kevin Hughes.

Customizations...

... are made in the configuration file, which we often rename *swish.sitename.config*.
Inside the configuration file, we specify that the *swish* index file is named *in-*

dex.sitename.swish. To shorten the length of nightly e-mail from the *cron* daemon, we set the value of *IndexReport* to 1, which provides minimal confirmation that the indexing worked. Finally, we add the words *internal* and *rcs* to the *FileRules* pathname criteria, to avoid indexing revision control information or proprietary ‘internal’ areas of the site.

```
IndexFile index.kensho.swish
# This is what the generated index file will be.

IndexReport 1
# This is how detailed you want reporting. You can specify numbers

FileRules pathname contains internal rcs admin testing demo trash confidential
# Files matching the above criteria will *not* be indexed.
```

cgi searching of *swish* indices via *wwwais* takes some time to set up, since there is a lot of directory to URL mapping that must be set up to perfection. Be patient!

Local Sources

Not usually stored on-site, since the customization takes place in the configuration files.

Official Distribution

<http://www.eit.com/software/swish/>

<ftp://ftp.eit.com/pub/web.software/swish/>

Official Documentation

<http://www.eit.com/software/swish/swish.html>

4.1.7 *wwwwais*

wwwwais is a cgi program that generates a W^3 search interface to *swish*, *wais*, and other full-text indices. *wwwwais* also formats and manages the search results. Guess who developed this program? Another Kevin Hughes/EIT resource.

Customizations

Compiling *wwwwais* from scratch requires one hard-coded file location to specify the configuration file, the rest can be set dynamically.

```
/* You will need to define the following option! */
#define CONFFILE "/usr/local/httpd/conf/wwwwais.conf"
/* The configuration file for wwwwais that holds all the defaults.*/
```

Local Sources

Not usually stored on-site, since the customization takes place in the configuration files.

Official Distribution

<http://www.eit.com/software/wwwwais/>

<ftp://ftp.eit.com/pub/web.software/wwwwais/>

Official Documentation

<http://www.eit.com/software/wwwais/wwwais.html>

4.2 W^3 Content Administration

4.2.1 *giftrans*

giftrans is a command-line tool to manipulate gif format images. It can convert conventional gif 87 format images to transparent, interlaced gif 89 format images. *giftrans* is used by the *getgraph.pl* script we run overnight on some sites. *giftrans* is also helpful for improving client-provided image files that were uploaded via ftp, without requiring additional file transfer.

Customizations

None.

Official Distribution

<ftp://ftp.rz.uni-karlsruhe.de/pub/net/www/tools/giftrans.c>

Official Documentation

<ftp://ftp.rz.uni-karlsruhe.de/pub/net/www/tools/giftrans.1>

4.2.2 *weblint*

weblint is a perl tool by Neil Bowers neilb@khoral.com. We use *weblint* to assure our HTML files comply with standard (or semi-standard) HTML specifications.

Customizations

None.

Local Sources

~/bin/weblint

Official Distribution

<ftp://ftp.khoral.com/pub/weblint/>

Official Documentation

<http://www.khoral.com/staff/neilb/weblint/manpage.html>

4.2.3 *rtfohtml* and *rtfoweb*

These two tools do a very practical job of converting rich-text-format word processor documents into completely hyper-linked HTML document trees. The current distribution is freeware, soon to become shareware.

A good summary of the two tools is offered on the W^3 distribution/documentation page:

rtfoweb extends *rtfohtml* by a few additional functions such as automatic splitting of the document, inserting a navigation panel into the HTML files and generating an index with active links. Together these tools provide quite a comfortable and yet

powerful means to get your (Word or whatever) documents on the Web.

Customizations

None.

Local Sources

Not stored on-site.

Official Distribution

<http://sun1.rrzn-user.uni-hannover.de/~zzhibol/rftoweb/guide-Title.html>

Official Documentation

<http://sun1.rrzn-user.uni-hannover.de/~zzhibol/rftoweb/guide-Title.html>

4.3 W^3 Access

4.3.1 *lynx*

lynx is the leading text-mode W^3 browser. We use it to check our user interfaces— to see what text-mode visitors see, and make sure key site functions are available to text-only users.

Customizations

None, we've only used binary distribution to date.

Local Sources

None.

Official Distribution

<ftp://ftp2.cc.ukans.edu/pub/lynx/>

Official Documentation

http://www.cc.ukans.edu/lynx_help/Lynx_users_guide.html

Unofficial Documentation

<http://www.nyu.edu/pages/wsn/subir/lynx.html>

4.3.2 NCSA *Mosaic*

The most popular public-domain graphical W^3 browser. We use it to check our user interfaces. The *autosurf* feature is useful for testing complete W^3 site linkage. The *kiosk* and *presentation* modes also have occasional special applications.

Customizations

Default Font: Times New Roman 10pt
Default <PRE> Font: Lucida Console 8pt

Local Sources

Mosaic source is not in the public domain

Official Distribution

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html> ■
<ftp://ftp.ncsa.uiuc.edu/Web/Mosaic/>

Official Documentation

<http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html> ■
<http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/Draft/index.htm> ■

4.3.3 Netscape Navigator

80% of W^3 users now see the World Wide Web through the user interface of Netscape Navigator W^3 browser software. Netscape Browsers on Win32, Macintosh, and UNIX platforms are used to design and test all of our Kensho sites and user interfaces.

Customizations

Default Font: Times New Roman 10pt

Default <PRE> Font: Lucida Console 8pt

Official Distribution

<http://www.netscape.com/comprod/mirror/index.html>

Official Documentation

<http://home.netscape.com/assist/support/client/index.html>

4.4 Internet Utilities

4.4.1 *telnet* and *rlogin*

These tools were described in detail earlier, here are the URL's you'll need to retrieve the software:

Macintosh: NCSA *telnet*

From the University of Illinois at Urbana–Champaign

<http://www.ncsa.uiuc.edu/SDG/Software/Brochure/MacSoftDesc.html#MacTelnet>

Windows: *QVT/Net*

<ftp://biocserver.bioc.cwru.edu/pub/windows/qvtnet/>

4.4.2 *ftp*

These tools were described in detail earlier, here are the URL's you'll need to retrieve the software:

Macintosh: *Fetch*

<http://www.dartmouth.edu/pages/softdev/fetch.html>

UNIX: *ncftp*

<ftp://ftp.cs.unl.edu/pub/ncftp/>

Windows: *ws_ftp*

<http://www.csra.net/junodj/>

<ftp://FTP.USMA.EDU/pub/msdos/winsock.files/>

4.5 Electronic Mail

4.5.1 *hypermail*

hypermail is an excellent tool for archiving e-mail messages in a format accessible via the *W*³. We've used *hypermail* to index both public and private e-mail lists with great success. Score one more for Kevin Hughes of EIT, he developed *hypermail* too.

Customizations

Depending on how *hypermail* is installed, it will require modification of an account's *.forward* or *.procmailrc* file to route the appropriate incoming messages to the *hypermail* program for processing.

Local Sources

Not stored on-site.

Official Distribution

<http://www.eit.com/software/hypermail/>

<ftp://ftp.eit.com/pub/web.software/hypermail/>

Official Documentation

<http://www.eit.com/software/hypermail/hypermail.html>

4.5.2 *procmail*

The ultimate e-mail processing application, from Stephen R. van den Berg at RWTH-Aachen, Germany berg@pool.informatik.rwth-aachen.de. *procmail* can turn a single user UNIX account into a decent resource to process e-mail for an entire MX domain.

Customizations

Usually self-configures with an elaborate, interactive installation script.

Local Sources

Not stored on-site.

Official Distribution

<ftp://ftp.informatik.rwth-aachen.de/pub/packages/procmail/>

Semi-Official Documentation

<http://www.tkg.com/people/curt/procmail.html>

Unofficial Documentation

<http://www.jazzie.com/ii/internet/procmail/>

<http://www.gl.umbc.edu/~ian/procmail.html>

4.5.3 *smartlist*

The *procm*ail-based mail list management application, from *procm*ail author Stephen R. van den Berg at RWTH–Aachen, Germany berg@pool.informatik.rwth-aachen.de. *smartlist* outperforms other list management applications such as *majordomo*, offers more sophisticated features, and can be run from a virtual e–mail domain.

Customizations

Usually self–configures with an elaborate, interactive installation script. Integrates with an existing *procm*ail installation. The text file *Manual*, installed in the `~smartlist/.etc/` subdirectory, is an excellent manual for the program.

Local Sources

`~smartlist`

Official Distribution

[`ftp://ftp.informatik.rwth-aachen.de/pub/packages/procm/`](ftp://ftp.informatik.rwth-aachen.de/pub/packages/procm/)

Official Documentation

[`http://www.inovadx.com/internal/Manual`](http://www.inovadx.com/internal/Manual)

`\~{}/smartlist/.etc/Manual`

4.6 General

4.6.1 GNU Utilities

The Gnu's Not UNIX (GNU) Utilities are collaboratively developed by volunteers all over the Internet. All GNU software is released into the public domain, protected (copylefted) by the Free Software Foundation's GNU Public License (GPL). The GNU Project has created improved replacements for a number of standard UNIX utilities. In addition, many essential software development tools are created and maintained by the GNU Project.

cvs

The Concurrent Versions System (*cvs*) is an extension of the gnu Revision Control System. *cvs* allows project-level version management, rather than file-by-file. We're presently evaluating *cvs* for W^3 site design and maintenance.

Customizations None.

Local Sources Not stored on site.

Official Distribution <http://www.cyclic.com/cyclic-pages/getting.html>

Semi-Official Documentation <http://www.loria.fr/~molli/cvs-index.html>
http://www.omnigroup.com/Documentation/cvs/cvs_toc.html

gzip

This utility was described in detail earlier, here's the URL you'll need to retrieve the software:

<ftp://prep.ai.mit.edu/pub/gnu/>

joe

This text editor was introduced in detail earlier, here are the URL's you'll need to retrieve it:

<http://www.srce.hr/~hniksic/computers/joe.html>

<ftp://ftp.std.com/src/editors/>

rcs

GNU's Revision Control System (*rcs*) is used to maintain revision-control information of source files, text files, binaries, and as we employ it, HTML files. By managing W^3 site development with GNU *rcs*, we are able to safely turn several people loose to develop the same site. We can safely revert to old versions when changes don't meet client approval. Here's a summary from the *rcsintro(1)* manpage:

The Revision Control System (RCS) manages multiple revisions of files. RCS automates the storing, retrieval, logging, identification, and merging of revisions. RCS is useful for text that is revised frequently, for example programs, documentation, graphics, papers, and form letters.

Customizations None. Compiles out of the box.

Local Sources Not stored on-site.

Official Distribution <ftp://cs.purdue.edu/pub/RCS/>

Official Documentation <http://www.ai.mit.edu:80/info/rcs/!!first>

Unofficial Documentation <http://www.iac.honeywell.com/Pub/Tech/CM/CMFAQ.html>
<http://www.cs.hmc.edu/FAQ/qref/rcs.html>

tar

GNU *tar* offers many enhancements over the *tar* binary supplied by commercial vendors such as Sun and SGI. Here's where to grab the source:

<ftp://prep.ai.mit.edu/pub/gnu/>

4.7 Helpful Resources

The resources identified here should help you stay in tune with the rapidly evolving Internet *W*³ community—don't hesitate to set aside 20–30 minutes a day to participate in some UseNet discussions that are relevant to your work!

4.7.1 HTML Standard

Hyper Text Markup Language (HTML) is a subset of Standard General Markup Language (sgml). Both 'languages' are standardized rules for page description. The standards development process is ongoing, and it is helpful to be familiar with the latest HTML standard and proposed extensions. The standards development is coordinated by the *W*³ organization. The standards documents are available in HTML format (of course!). These HTML specifications are especially useful each time new browser features become available.

Customizations

N/A

Local Sources

Not stored on-line.

Official Distribution

<http://www.w3.org/pub/WWW/MarkUp/>

Official Documentation

<http://www.w3.org/pub/WWW/MarkUp/>

4.7.2 CGI Standard

The Common Gateway Interface (cgi) standard was developed to extend the World Wide Web beyond the limited built-in feature set offered by http and HTML. Most 'behind-the-scenes' customizations of W^3 sites are implemented as CGI applications—online payment systems, order forms, tests, custom URL tracking, etc. The NCSA Team maintains documentation, tutorials, and related links to promote the cgi.

Customizations

N/A

Local Sources

Not stored on-site.

Official Distribution

<http://hoohoo.ncsa.uiuc.edu/cgi/>

Official Documentation

<http://hoohoo.ncsa.uiuc.edu/cgi/>

4.7.3 Mailing List Management Software FAQ

In situations where *smartlist* is not available, or when you or a client need to weigh the pros and cons of the various mail list management tools, there is an excellent Frequently Asked Questions document available online:

<http://www.cis.ohio-state.edu/hypertext/faq/usenet/mail/list-admin/software-faq/faq.html>

4.7.4 W^3 Newsgroups

There are more than twenty UseNet newsgroups whose discussion centers around the World Wide Web. They are an excellent source of current information and collective assistance. The charters of all these groups can be browsed on the W^3 at:

<http://boutell.com/%7Egrant/web-groups.html>

Here is a summary table derived from Thomas Boutell's online directory:

Table 4.1: W^3 -related UseNet Newsgroups

W3 Newsgroups

GENERAL

comp.infosystems.www.advocacy Comments and arguments over the best and worst.

comp.infosystems.www.announce World-Wide Web announcements. (Moderated)

comp.infosystems.www.misc Miscellaneous World Wide Web discussion.

BROWSERS

comp.infosystems.www.browsers.mac Web browsers for the Macintosh platform.

comp.infosystems.www.browsers.misc Web browsers for other platforms.

comp.infosystems.www.browsers.ms-windows Web browsers for MS Windows.

comp.infosystems.www.browsers.x Web browsers for the X-Window system.

SERVERS

comp.infosystems.www.servers.ms-windows Web servers for MS Windows and NT.

comp.infosystems.www.servers.unix Web servers for UNIX platforms.

comp.infosystems.www.servers.mac Web servers for the Macintosh platform.

comp.infosystems.www.servers.misc Web servers for other platforms.

AUTHORING

comp.infosystems.www.authoring.cgi Writing CGI scripts for the Web.

comp.infosystems.www.authoring.html Writing HTML for the Web.

comp.infosystems.www.authoring.images Using images, imagemaps on the Web.

comp.infosystems.www.authoring.misc Miscellaneous Web authoring issues.

 W^3 RELATED

alt.culture.www Web Culture

alt.hypertext Hypertext

alt.lang.vrml Virtual Reality Modeling Language (vrml)

alt.www.hotjava java

comp.lang.java java

comp.lang.javascript JavaScript

comp.text.sgml Standard General Markup Language (sgml)

Chapter 5

Endnotes

1 Sobell, Mark G. A Practical Guide to the UNIX System. Redwood City, California: Benjamin/Cummings Publishing Company, Inc., 1995.

2 Frisch, Aleen. Essential System Administration. Sebastopol, CA: O'Reilly & Associates, c1991.

3 Shiffrin, Richard M.; Nosofsky, Robert M. "Seven Plus Or Minus Two: A Commentary On Capacity Limitations." Psychological Review v101, n2 (April, 1994):357.

4 Bates, Marcia J. (1989). "The Design Of Browsing And Berrypicking Techniques For The Online Search Interface," Online Review, 13(5), p. 407–24.

5 Shneiderman, Ben. Designing The User Interface : Strategies For Effective Human–Computer Interaction. Reading, Mass.: Addison–Wesley, c1987.

6 Liu, Cricket, et.al. Managing Internet Information Services. Sebastopol, California: O'Reilly & Associates, Inc., 1994.

Chapter 6

Index

A archiving/compression software. See compression/archiving software ARK Enterprises 71 at job scheduling utility. See cron job scheduling utility

B Berg, S.R. van den 51 berrypicking 16 Billimoria, Neville 7 binary formats 20 Boutell, Thomas 96

C capitalization. See case sensitivity case sensitivity 19 cgi standard 94 chunking 16 collocation 16 compression/archiving software 37 gnu zip 37, 37-39, 39, 40, 41, 41-46, 90-91 other formats 41-46 tar 34, 37, 37-39, 38-39, 39, 40, 40-41, 41, 41-46, 77, 91-97 Concurrent Version System (cvs) 90. See also revision control cron job scheduling utility 42-43, 43, 80-81 at as an alternative to cron 43 crontab 42-43, 43, 43-74 using with shell scripts 42-43 cvs. See Concurrent Version System (cvs) cyclelog 76. See also getgraph

D design 3, 6, 7, 10, 16, 18, 19, 67, 85, 90 background 18, 73 concepts 18 physical size 18 visual impact 18 Dreilinger, Sean 7

E e-mail mua, eudora 57 user-agent configuration 57

F file formats 17, 72, 73 file naming considerations by os 20 File Transfer Protocol (ftp) 23 using with Macintosh 35, 86 using with MS-Windows 35, 86 using with UNIX 34, 41, 86 using with *W*³ browsers 36 filename length 19, 20 formatting 4

G getgraph 77, 78, 82 getstats 10, 20, 78 graphics Adobe Photoshop 72 Aldus Photostyler 72 gif file format 73 *giftrans* 82 imagemaps 12, 72, 74 JASC Paint Shop Pro 72 jpeg file format 73 MapTHIS! 72 Micrografx Picture Publisher 72 png file format 73

H hardware 9, 48, 59, 73 history 7 HTML standard 32, 93 httpd, ncsa 79 httpd, Netscape 79 Hughes, Kevin 20, 45, 52, 78, 80, 81, 87 *hypermail* 52, 87 hypertext 17, 18, 95 Three golden rules of hypertext 17

I indexing 17, 42, 44, 45, 67, 68, 69, 80 key words 68 listing services 70 search engines 9, 67 search engines, Alta Vista 70 search engines, Comprehensive list of sites 70 search engines, Excite 70 search engines, InfoSeek (Directory and Guide) 70 search engines, InkTomi 70 search engines, Lycos 70 search engines, Open Text 70 search engines, WebCrawler 70 search engines, WWW Worm 70 search engines, Yahoo 70 information-seeking behavior serendipity 67 intelligent agents 3 Internet Service Providers (ISP's) 7, 8, 9, 26, 28, 43, 55, 59 BEST 64 information 7 newsgroups 28

J joe text editor for UNIX 31, 32, 63, 90

K Kensho Online Site Administration Guide 3 support 7

L laptop 9, 19 *lynx* 84

M Mail List Management access control 46 questions to ask yourself 46 size of list 46 mailing list management 95 memory. See chunking Microsoft Windows 95 72 Microsoft Windows NT 20, 72 Miller's 'chunking' findings 16 mosaic, ncsa 85

N Netscape navigator 85 networking 5 Neville Billimoria. See Billimoria, Neville newsgroups, *W*³-related 96

P Picasso, Pablo 3 pkzip 41 *procm* 59, 62 documentation available as manpages 62 recipes 48, 55, 59, 62 recipes, automatic responses 60 recipes, forward to individual 59 recipes, forward to list 60 publicizing *W*³ sites 69 advertising 71 information gatekeepers 70, 71 polite plugging 70 selected appeals 70 strategic signatures 70 Publicly Accessible Mailing Lists (paml) 52

R recall. See chunking reciprocal links 16, 70 revision control 20, 26, 31, 33, 61, 66 check in 33 *rcs* 33, 80, 91 *rlogin* 23, 86 *rtftohtml* 83 *rtftoweb* 83

S S.R. van den Berg. See Berg, S. R. van den satisficing 16 Sean Dreilinger 7 Shneiderman, Ben 17 Simple Mail Transfer Protocol (smtp) 55 *smartlist* 10, 28-29, 46-47, 47-48, 48-49, 49-50, 50-51, 51, 54-55, 60-61, 89, 95-96 createlist 47 custom list configuration file *rc.custom* 47, 49 custom subscription message 47, 49, 50 global initialization file 47, 48 removelist 54 Stebens, Alan 62 *swish* (Simple Web Indexing for Humans) 44, 45, 80, 81 *swish -c* 44

T

Table of Contents 18, 37, 38, 40, 78 Tasks adding or changing graphics 72 change an existing HTML page 31, 61 change the ‘automatic overnight’ stuff 42 connect to your website for remote administration 23 create an electronic mail discussion list 46 getting attention 16, 17, 67 transfer files to and from your site 34 unix housekeeping 26 *telnet* 23, 24, 25, 29, 31, 86 using with Macintosh 24 using with MS-Windows 25 using with UNIX 23 *telnet* and *rlogin* 23 Thomas Boutell. See Boutell, Thomas

U unix 5, 8, 19, 23, 23-24, 24, 26-27, 28-29, 29-30, 30, 32, 34, 37, 40, 41, 42, 47-48, 51, 59, 60-61, 63, 79, 85, 86, 88, 90 *bash* 29, 30 disk space 27 e-mail basics in the shell 27 filename completion 26 housekeeping/review 26 *motd* (message of the day) 27, 28 quota -v 28 shell configuration 26, 29, 30

tcsh 29 tildes 26, 61 UseNet News 28, 53, 70, 92, 96 using with Macintosh 28
using with MS-Windows 28 using with UNIX 28 user accounts 7, 8, 15, 24,
29, 36, 43, 49, 51, 52, 55, 59, 63, 66 user interface design 16 user interfaces
Five factors for evaluating user interfaces 17

V vi editor for UNIX 32

W *W*³ Access Control 63 access file 63 adding users 66 dbm user files 66
htpasswd 64, 65, 66 *weblint* 32, 33, 82 Wilson, Todd C. 72 *wwwais* 80, 81